

版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。

618

H E W A Y W E W I N

决战618

探秘京东技术 取胜之道

京东集团618作战指挥中心 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

京东集团618作战指挥中心

这里汇集着数百位京东最顶尖的研发人才，是京东最宝贵的财富。他们都是来自于京东各体系的技术大牛，既有核心系统的架构师，又有扑在一线的运维专家，既有奋战在科学前沿的研究型学者，又有在人工智能、大数据应用方面有一定建树的科学家。他们平时分散在各个部门，是各个部门的中坚力量，每到吹响618战斗号角的时候，他们会不约而同地聚集在这里，确保流量洪峰来临时系统安全、稳定、可靠，并提供最佳的用户体验。本书就是他们的扛鼎之作。



关注微信，发送“618”
加入本书共读群



京东 京东商业技术丛书

内容简介

本书围绕618战役复盘分析，结合京东618战役实战经验，从战略、战术、技术、运营、物流、客服、供应链、品牌、公关、法务、财务、人力资源、风险管理、信息安全、合规、社会责任、可持续发展等方面，全面剖析京东618战役的成功之道。本书可作为京东集团内部培训教材，也可作为其他企业学习借鉴。本书由京东集团618战役指挥部编著，京东集团618战役指挥部是京东集团618战役的总指挥部，负责统筹协调618战役的各项事宜。本书是京东集团618战役指挥部多年经验的结晶，也是京东集团618战役指挥部智慧的结晶。本书可作为京东集团内部培训教材，也可作为其他企业学习借鉴。本书由京东集团618战役指挥部编著，京东集团618战役指挥部是京东集团618战役的总指挥部，负责统筹协调618战役的各项事宜。本书是京东集团618战役指挥部多年经验的结晶，也是京东集团618战役指挥部智慧的结晶。

决战618

探秘京东技术 取胜之道

京东集团618作战指挥中心 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以京东技术团队备战618为主线,集合京东数百位技术专家,对京东所有和618相关的关键技术系统进行了一次全面的梳理和总结,是京东技术体系的智慧结晶。

本书从前端的网站、移动入口到后端的结算、履约、物流、供应链等体系,系统展示了京东最新的技术成就。同时,也涵盖了京东正在充分运用大数据、人工智能等先进技术对所有技术体系架构进行整体改造,使其始终保障技术的先进性的方法,以及京东对未来科技发展的积极探索和展望。

618盛况往往意味着汹涌的流量洪峰,京东技术团队如何保证其背后庞杂系统的安全、可靠与稳定,如何确保系统的高可用性、高并发性、高可扩展性,如果您对此有兴趣,请随本书一探究竟吧!

本书可供广大互联网技术人员和商业从业者阅读,也可供科研人员和高校相关专业的师生参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

决战618:探秘京东技术取胜之道/京东集团618作战指挥中心编著.

北京:电子工业出版社,2017.11

(京东商业技术丛书)

ISBN 978-7-121-32730-8

I. ①决… II. ①京… III. ①电子商务-网站-应用程序-程序设计 IV. ①F713.361.2②TP393.092

中国版本图书馆CIP数据核字(2017)第230974号

策划编辑:孙学瑛 康 旭 宋亚东

责任编辑:孙学瑛

印 刷:北京捷迅佳彩印刷有限公司

装 订:北京捷迅佳彩印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:787×1092 1/16 印张:21.75 字数:543千字

版 次:2017年11月第1版

印 次:2017年11月第1次印刷

定 价:99.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至zltz@phei.com.cn,盗版侵权举报请发邮件至dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819,faq@phei.com.cn。

编委会

特邀顾问

京东集团CTO

张晨

京东集团副总裁（按姓氏拼音排序）

曹鹏 Eric Zhao 何刚 黎科峰 梁勇鸿 刘轶 Tony Lee 翁志 肖军

许凌 于永利 颜伟鹏

主编

马松

副主编

杨佼姣 王先科 杨庆广 贺津朝

编委会委员（按姓氏拼音排序）

白圣培 白贤锋 董钊 范超 高玮 韩笑跃 黄燃 胡景贺 李星毅 李峥

李轶 林世洪 刘海锋 刘刚 王梓晨 王海旭 肖军超 余文虎 赵湘建 张卉

特别感谢（按姓氏拼音排序）

程岩 陈伟 付彩宝 何方 贾东胜 李仑 李运鹏 李军亮 刘晖 刘会

刘彦伟 钱璟 尚鑫 王彪 王晓钟 王晓烽 王大泳 王彦明 吴迪 谢翔

熊健南 熊道辉 杨坤 杨瞳 杨静 杨超 臧志 者文明 朱艳波 搜索深

圳团队



参编（按姓氏拼音排序）

安博文	安 宁	安 培	安 涛	包勇军	鲍永成	卞 祥	蔡 想	陈保安	陈 诚
陈洪涛	陈 磊	陈 龙	陈 璐	陈乾龙	陈全峰	陈英杰	陈月华	陈中兵	程晓磊
邓 波	邓 聪	董 浩	董鲁北	杜 亮	杜庆鸿	杜晓玉	樊建刚	冯 成	冯春宇
冯捷春	冯 军	冯俊杰	冯 亮	冯 卓	盖美红	高 飞	高 玮	耿 爽	宫敬财
龚雪平	关攀攀	郭 凝	郭文涛	郭 欣	郭 鑫	郭玉锐	郭智林	韩嘉星	韩贞阳
何红伟	何文斌	何晓强	贺松林	贺占领	贺长荣	赫 南	赫 阳	胡宝彬	胡 峰
胡 浩	胡 伟	扈克然	花 璞	华家乐	黄 嘉	黄明源	黄 平	黄 伟	黄肖明
黄小斌	黄 轩	季 健	贾瑞卿	江 军	江 雪	姜友德	蒋佳涛	蒋瑞祺	焦文健
金 洪	敬疆川	兰华勇	李 闯	李春光	李頔璇	李东耕	李 凡	李凤娇	李冠权
李海龙	李慧岩	李建奇	李金萍	李军亮	李科君	李 猛	李思倩	李晓玲	李秀海
李祎翔	李 瑛	李友科	李振军	李 政	李尊敬	梁洪成	梁小平	梁晓飞	梁增海
廖 承	廖 鹏	林金明	林伟腾	林战刚	刘贝贝	刘 丹	刘广煜	刘建平	刘 健
刘江斌	刘 静	刘峻桦	刘 林	刘朋飞	刘 鹏	刘文冈	刘吓凤	刘 旭	刘亚超
刘 彧	刘 诏	罗 超	罗 金	罗重文	马 凯	马思宁	马 欣	梅志文	孟妍妍
米中华	苗延超	莫飘萍	聂启忠	聂学贤	牛红振	潘小光	彭笔立	彭正军	钱 璟
乔晓强	秦筱桦	邱俊傑	邱全涛	全金凤	任莉强	闰秋胜	沈 晨	沈 亮	寿如阳
宋 磊	孙 彬	孙海滨	孙红静	覃朝光	唐 亮	滕伟伟	屠志强	汪宝杰	汪 瑶
王春兰	王代起	王丰华	王海涛	王建磊	王 晶	王丽君	王利辉	王 宁	王顺达
王 涛	王 威	王 伟	王晓烽	王新栋	王雅瑾	王妍革	王 洋	王 永	王宇瑄
王 禹	王 玉	王治澎	王祖文	韦 仕	魏仁言	吴定东	吴昊挺	吴 鹄	吴 桐
吴先远	武 磊	武 林	息 宾	夏庆峰	夏先刚	向 彪	肖明昭	肖 维	肖 雄
肖智鹏	谢 幕	邢嵩林	徐 峰	徐 奉	徐 军	徐开廷	徐 梅	徐 森	徐夙龙
徐文胜	徐新坤	徐莹雪	许 鹏	许 震	薛高飞	薛 韬	杨 超	杨大利	杨东洋
杨建强	杨丽敏	杨少利	杨泽森	杨 震	叶长根	易重伟	阴培培	尹德位	尤凤凯
于林坤	于明辉	于 绮	余 浩	余 珽	袁建伟	咎秀敏	张春蕾	张冬盟	张栋芳
张海兵	张洪光	张俊俊	张 侃	张克房	张 磊	张 露	张启祥	张 强	张树文
张 帅	张 伟	张永峰	张泽华	张子敬	赵 波	赵宏波	赵苗苗	赵伟柏	赵贤义
赵玉开	赵云霄	赵紫星	者文明	周吉鑫	周敬岩	周琪力	周 涛	周 旭	周银华
周 正	周治平	朱厚强	朱月飞	朱志国	祝旭晖	邹 毅			

序

T H E W A Y W E W I N

技术引领 开创京东新未来

2017年的开年大会，我给所有京东人做了一场不太一样的演讲，主题只有一个——技术！

从写下京东的第一行代码开始，到看着它一步步成长为先进技术、架构、模式的领先应用者，京东技术的飞跃发展让我感慨万千。而京东技术的大考，便是每年一次的“决战618”。

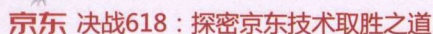
回首十几年前，京东的618都是在焦灼中度过的。一方面，我们期待大量的用户能参与到我们的促销中来；另一方面，我们又担心网站是否能够承受这样的访问量，是否会宕机……而如今，618已从京东店庆日发展成为了一场全民参与的消费狂欢和行业盛会，在618作战指挥中心的大屏幕上，不断攀升的曲线，跳跃的数字，都让我们充满信心和自豪感。

这是所有京东技术人打造的硬实力，《决战618：探秘京东技术取胜之道》正是对于京东过去十三年间技术整体发展的总结和盘点，代表了京东已经达到的出色技术成就。

着眼未来，时代正在发生剧烈的变化，科技的进步速度会越来越快，科技对人类、对企业产生的影响将超过过去100年，科技对我们每一个人、每家公司、每个企业，甚至学校、非政府组织，都会带来巨大的改变。

但零售的本质从未发生改变——依旧是成本、效率和用户体验！未来，京东将全面地向技术转型，让技术从被动应对变为主动引领，持续优化成本、效率和用户体验，打造一个包括智能商业、智能金融、智能保险业务在内，全球领先的智能商业体。

此前，我们用了很长时间让大家认同京东是一家非常成功的零售公司，我坚信，接下来我们会用同样长甚至更短的时间，让大家认同京东是一家非常成功的技术公司。创新一直都是京东人弥足珍贵的品质和锐意进取的动力。公司也将从文化重塑、组织结构、激励方式等方面寻求巨大的改变，给予创新以包容和试错，保障技术转型的全面落地。



T H E W A Y W E W I N

京东研发的兄弟们，你们一定可以做到！

京东集团董事局主席兼首席执行官

序

T H E W A Y W E W I N

二

决战 618 背后的技术实质，是未来的人工智能

提到618，消费者会想到大量的促销，琳琅满目的商品，对于技术人，则会立刻联想到海量访问和订单，以及对整个系统的压力。当这本样书送到我的手中时，我第一个想法就是——这一定是本涵盖各种高抗压架构、应急预案管理的“IT书”；但仔细读下来，我却欣喜地看到大量人工智能的应用。在“决战618”背后的关键技术中，是未来的人工智能。

人工智能无疑是当前最为热门的技术方向，在一些领域取得了初步的成就。而人工智能应用要想更快速地发展，则需要依赖于大数据的收集、挖掘和应用。京东作为中国线上线下最大的零售商，完整覆盖从采购、售卖到配送、客服等全电商流程，拥有中国电商领域最完整、最精准、价值链最长的数据，为人工智能的充分实践提供了丰富的数据基础。这些数据为京东发展人工智能的应用提供了肥沃的土壤。基于这样的数据，人工智能可以用来分析用户需求，进行自动定价，自动补货，可以支持无人仓储，也可以利用无人车和无人机送货。在书中，我们可以看到京东通过人工智能的研发与实践也在不断地推动着电商的革新。

例如，在图像识别、语音识别方面，商业上已经有了很多很好的应用案例，今天我们都是这些应用的享用者及受益者。拿京东的JIMI人工智能客服机器人来说，其在机器学习、深度神经网络、知识图谱构建、自然语言处理方面均有较深的研究和分享，这一部分内容在本书第5.3节中有很好的梳理和分享。

但是同时我们也在思考，这些人工智能的成就到底得益于哪些条件呢？为什么这些成就在十年或者二十年以前却没有出现呢？如何能够去预测一项人工智能技术，是不是能够产生它所预期的作用？

仔细读这本书，会发现问题迎刃而解。读下来，你会发现，书中各个章节的分享都离不开人工智能的技术，如智慧供应链通过大数据分析 with 深度学习建模，从提供给京东采销内部使用到现在成为开放给外部供应商的基础设施；不同于传统的人工智能项目中对于广告的玩法、面对个性化的推荐应用，京东如何对复杂问题提取特征，建模和高效训练，做到在很短的时间内对每一次广告、推荐请求



THE WAY WE WIN

作出毫秒级预估服务。可以看到，这些都是基于京东清晰的商业模式需求、高质量反馈的大数据、清晰的问题定义和使用领域边界，以及擅长算法的、跨界的人工智能人才。而更让我兴奋的是，以618为代表的大量应用场景和运营产生了大量的精准数据积累，也为人工智能的应用铺平了道路。

从人工智能的长远发展来看，我一直觉得基于实验室的研究工作和接触大规模复杂的工业问题是不可分割的。在人工智能的研究中，在实验室工作、参加学术会议或者拜访一些其他的实验室，可以关注整个业界的最新发展；而在工业界，可以大规模地利用人工智能来解决实际应用上的痛点问题。这本书告诉你，京东，是工业界人工智能应用的一个典范，通读此书，想必会受益颇多。

杨强

香港科技大学计算机系主任及大数据研究院院长、教授

序 三

T H E W A Y W E W I N

618，打响技术攻坚战

京东618始于京东的店庆日，现在早已演变成为全民参与的网购狂欢节。2017年6月18日，当京东总部的618战报大屏定格在“当前累计下单金额1199亿元”时，欢呼声、掌声响彻整个作战指挥室。在成绩背后，充分体现出京东的技术硬能力。对于京东技术人来讲，每一年的618都是一场巨大的考验，而在这样一个考验面前，从来就不允许失败。

作为京东商城研发体系的负责人，2012年临危受命，成为京东618的技术总指挥，在此后5年中，我带领技术团队，发起了一场又一场技术攻坚战，见证了京东618如何促进技术发展、推动技术创新、激励团队协作。作为技术老兵，我想告诉大家，京东618不只是一次购物节，更是一场战斗，一场弥漫着硝烟的战斗。

2012年6月18日，随着业务的高速发展，京东的订单量骤增。当时的内部业务支撑系统已经达到上千个，作为一个靠快速迭代出来的系统，我们的系统架构、服务框架、质量控制体系等都没有做好准备。618当天，在面对瞬时访问峰值骤增的情况下，一些系统的服务质量严重下降，导致POP商家看不到新的订单信息，造成商家无法履约等严重问题。

架构升级迫在眉睫，统筹规划也势在必行，618技术团队开启系统升级、团队协作的作战之路，系统每年都有更新、演变。2013年，京东的交易系统开启了去Oracle、去SQL Server行动，减少对商业数据库的依赖，引入开源可控的数据库，提升系统的稳定性；自主研发JFS、JMDB、JMQ、JSF等中间件和分布式存储系统，支撑全京东的无数业务。2014年，建设指挥中心大屏，实现对核心销售、安全监控、性能监控、京东直播、运营生产等全面监控；2015年，完成多中心交易项目，交易数据的多中心化，大规模Docker化部署，极大提升了资源利用率和生产效率；2016年，率先引入了“对抗军演”，通过模拟故障制造和故障恢复，极大地检验了故障预案可行性，考验团队处理随机事件的响应能力。2017年，数据驱动备战，技术改变生活，为京东第四次零售革命奠定坚实的技术基础。通过前期的不懈努力，2015年后，618已变成技术团队对京东业务、对供应商、对商家、对用户展示技



THE WAY WE WIN

术硬实力的舞台。

618从不是一个人、一个团队、一个部门在战斗，而是所有的京东人都在参与，是一场需要多兵种协同作战的战斗。在2017年的开年大会上，京东集团董事局主席兼首席执行官刘强东先生为京东未来12年的发展奠定了基调，“技术、技术、技术”，京东未来将是一家技术公司，一家依靠技术创新进行商业发展的公司。同时，第四次零售革命的到来，给我们带来了机遇，也带来了全新的未知挑战。因此，打造一个懂得协同、能打硬仗的团队已经成为我们首要的任务，相信在不久的将来，大家会看到这个变化。

本书全面、客观地描述了京东618技术演进之路，从产品规划、技术实现的不同维度，阐述了各个系统的发展之路。不管是产品经理、程序员、测试工程师，还是网店店主、互联网其他从业人员等，都能找到自己感兴趣的内容。

阅读这本书，感觉又回到了那个熟悉的指挥中心。展示着战报的实时大屏，屏幕上跳动的数字，一起熬夜备战的兄弟们……一切都历历在目。在此，谢谢所有参与本书创作的同学们，是你们让京东618以另外一种方式呈现在读者面前，让读者重新认识了京东618，重新认识了京东技术人。

马松

京东集团高级副总裁

前言

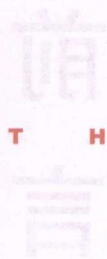
T H E W A Y W E W I N

过去，技术通常被看做是一切业务的基础，是支持业务发展的动力之一。通过技术可以把传统的业务流程数字化、信息化，让业务人员从事更具决策性或创造性的任务，所以技术常常被看作是一种工具。但是，随着人工智能、大数据、云计算的高速发展，大企业逐渐意识到技术不仅仅是工具，它可以驱动公司业务增长、提升用户体验、产生越来越多的创新力。特别是在互联网经济带来的天翻地覆的变革中，人们深刻感受到技术无疑是这次变革背后最大的推动力，企业要进一步发展，必须依靠技术领先与技术创新。因此，技术已成为各大互联网公司的必争之地。

作为这次互联网变革中的参与者与受益者，京东通过技术创新，在成本、效率、用户体验三大维度上取得了巨大成功。京东已由传统电商企业转变为一家以技术创新去改变传统零售、传统物流、传统金融乃至一切与用户体验相关的公司。在京东一次次地刷新618创造的天量交易额的同时，外界一次次为之震惊赞叹。众所周知，当企业的业务量达到一定规模，同时对发展速度有非常高的要求时，技术如何满足业务需求将会变为巨大的挑战。特别是像京东这样体量的互联网平台，在618期间面临的技术压力在整个IT技术领域也是比较罕见的。人们不禁想问，京东是如何做到的？为了与广大技术人员分享京东人在618备战中所做的技术保障工作以及积累的实践经验，推动行业技术进步，我们特推出此书。

本书是众多一线技术人员的智慧结晶，是无数团队历次攻坚克难的经验总结，更是所有参与备战的项目组人员的全面回顾与理性思考。本书基本囊括了京东所有的技术体系，是对业务保障方面技术的全面梳理，特别分享了京东在弹性云、多中心交易、高可用架构、平台化、风控系统等领域的实践经验。读者从中可以体会到，618的出色表现既得益于京东坚实的基础架构、稳妥的应急预案、成熟的调度指挥，又得益于先进的人工智能技术。

本书内容是理论与实战相结合，涉及的知识点较多、信息量较大，读者可以按顺序阅读每部分，也可以从中选取自己关心或感兴趣的章节阅读。



THE WAY WE WIN

感谢所有为本书做出贡献的京东研发、公关、图书等部门的同事。正是你们的全力支持和理解，利用节假日和休息时间加班加点写作、修改，本书才得以按时完成撰写与出版。本书整个的写作与出版过程也再次体现了京东人的高效和协作精神。同时，也特别感谢电子工业出版社博文视点的编辑们，正是你们的全力支持和配合，才得以让本书在不到两个月的时间出版。正是通过我们双方的努力，共同创造了出版奇迹。

由于时间仓促，书中难免有疏漏之处，敬请广大读者批评指正！

著者

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务：

提交勘误：您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。

与作者交流：在页面下方【读者评论】处留下您的疑问或观点，与作者和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/32730>



目录



第 1 章 京东 618 前世今生

1.1 京东 618 研发奋战之路 / 2

1.1.1 发轫之始（1998—2003 年） / 2

1.1.2 奠基之初（2004—2009 年） / 3

1.1.3 收获之期（2010—2016 年） / 3

1.1.4 未来已来（2017 年以后） / 5

1.2 京东 618 技术备战概览 / 6

1.2.1 技术备战工作重点 / 6

1.2.2 技术备战利器 / 7



第 2 章 万亿交易——丝滑平顺，坚如磐石 / 9

2.1 永不消失的网站 / 10

2.1.1 首页优化之路 / 10

2.1.2 商品详情页的架构变迁 / 12

2.1.3 分类列表探索之路 / 14

2.1.4 网站监控 / 16

2.2 交易平台 / 17

2.2.1 交易平台技术演进 / 18

2.2.2 交易平台架构揭秘 / 21

2.2.3 商品评价系统 / 24

2.2.4 质量控制 / 28

2.3 京东支付——更便捷，更省钱 / 31

2.3.1 移动支付 / 32

2.3.2 交易系统 / 33

2.3.3 自动网关 / 36

2.3.4 智能决策 / 38

2.3.5 支付未来 / 41



2.4 白条业务——先消费，后付款 / 42

2.4.1 智能授信 / 42

2.4.2 便捷交易 / 45

2.4.3 白条精细化营销 / 47



第 3 章 移动互联——便捷个性，体验至上 / 51

3.1 超级流量入口——App / 52

3.1.1 基础服务 / 52

3.1.2 技术支持 / 54

3.1.3 通天塔 / 57

3.1.4 秒杀系统 / 59

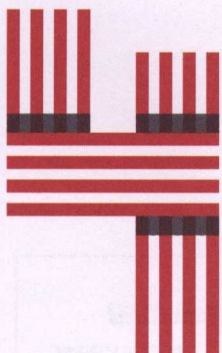
3.1.5 M 站 / 61

3.1.6 核心流程 / 62

3.2 渠道引流核武器——微信手 Q 购物 / 67

3.2.1 业务支撑修炼之道 / 67

3.2.2 技术运营发展之路 / 73



第 4 章 基础保障——坚实高效，安全稳固 / 81

4.1 基础架构 / 82

4.1.1 容器集群 / 82

4.1.2 数据库 / 85

4.1.3 中间件 / 87

4.1.4 智能运维 / 93

4.1.5 异地多活 / 97

4.2 天网系统——商城风控 / 100

4.2.1 发展历程 / 101

4.2.2 天网重点业务 / 101

4.2.3 天网系统架构 / 102

4.3 流量支撑平台 / 106

4.3.1 稳定压倒一切 / 106

- 4.3.2 基础网络 / 107
- 4.3.3 京东高可用前端接入系统 JFE / 110
- 4.3.4 京东 CDN / 113
- 4.3.5 防攻击系统 / 115
- 4.4 反刷单——为京东交易生态保驾护航 / 118
 - 4.4.1 反刷单系统架构 / 119
 - 4.4.2 特征工厂 / 119
 - 4.4.3 决策引擎 / 120
 - 4.4.4 下游管控 / 120
- 4.5 信息安全——重于泰山 / 121
 - 4.5.1 筑起安全的长城 / 121
 - 4.5.2 来自攻防第一线的博弈 / 125
 - 4.5.3 全民保障 618 / 128
- 4.6 揭秘京东大促数据屏墙 / 130
 - 4.6.1 系统概述 / 130
 - 4.6.2 系统实现 / 131
 - 4.6.3 系统价值 / 134
- 第 5 章 智能履约——瞬息万里，使命必达 / 135**
 - 5.1 智慧供应链 / 136
 - 5.1.1 智慧库存 / 136
 - 5.1.2 深度协同 / 140
 - 5.1.3 智慧定价 / 145
 - 5.1.4 智慧履约 / 148
 - 5.1.5 618 系统质量保障 / 154
 - 5.2 智慧物流 / 155
 - 5.2.1 精准履约与运营数据服务 / 155
 - 5.2.2 智慧仓储 / 161
 - 5.2.3 智能配送 / 165
 - 5.2.4 无人机、无人车和无人仓 / 174
 - 5.3 智能客服 / 181
 - 5.3.1 咚咚和 JIMI 历史演进 / 181
 - 5.3.2 产品形态与业务模式 / 182





5.3.3 京东咚咚和 JIMI 架构体系 / 183

5.4 智能财务 / 187

5.4.1 结算的秘密 / 187

5.4.2 你下单我算账，你不知道的来龙去脉 / 193

5.4.3 税控背后的故事 / 201

5.4.4 智能资金管控 / 205

第 6 章 赋能商家——生态共享，虚拟征程 / 209

6.1 开放平台营销系统 / 210

6.1.1 促销系统 / 210

6.1.2 订单系统 / 211

6.1.3 结算系统 / 213

6.1.4 商家系统 / 215

6.2 生态系统，保驾护航 / 219

6.2.1 京盾系统 / 219

6.2.2 正阳门系统 / 221

6.2.3 星盘系统 / 223

6.2.4 商智 / 225

6.2.5 京麦 / 229

6.3 虚拟平台的 618 征程 / 231

6.3.1 智能调度出口流量平台 / 231

6.3.2 虚拟业务黄金交易流程 / 233

6.3.3 京东户簿建设之路 / 236

6.3.4 虚拟业务的风控挑战 / 239

6.3.5 虚拟业务全渠道推进 / 241

6.3.6 质量保障与实战方案 / 243

6.4 创新业务 / 246

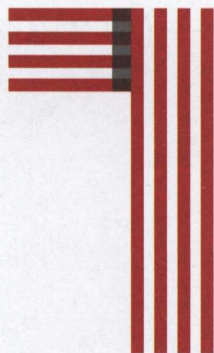
6.4.1 京东的“海外丝绸之路” / 246

6.4.2 营销产品技术的变革 / 249

6.4.3 京东 B2B 征程 / 254

6.4.4 重点创新电商业务 / 257





第 7 章 智慧营销——知你所想，予你所求 / 263

7.1 智慧营销 / 264

- 7.1.1 智慧营销平台的前世今生 / 264
- 7.1.2 促销七十二变 / 266
- 7.1.3 优惠券三十六计 / 268
- 7.1.4 购物助手 / 269
- 7.1.5 智慧营销生态化 / 271

7.2 智能广告 / 273

- 7.2.1 原生广告 / 273
- 7.2.2 站外流量赋能商家 / 276
- 7.2.3 PINO 人工智能 / 277
- 7.2.4 反作弊系统 / 278

7.3 精准搜索 / 281

- 7.3.1 搜索架构演进 / 282
- 7.3.2 分布式搜索引擎 / 283
- 7.3.3 数据索引 / 284
- 7.3.4 搜索系统的 DevOps 之路 / 285

7.4 智能推荐 / 287

- 7.4.1 推荐产品 / 287
- 7.4.2 推荐系统架构 / 288
- 7.4.3 数据平台 / 290
- 7.4.4 关键技术 / 291

7.5 玩转大数据 / 296

- 7.5.1 大数据综述 / 296
- 7.5.2 技术平台 / 297
- 7.5.3 数据仓库 / 303
- 7.5.4 数据应用 / 307



第 8 章 项目管理——运筹帷幄，万众一心 / 313

8.1 618 项目管理挑战 / 314

- 8.1.1 项目管理的难度 / 314
- 8.1.2 项目经理能力要求 / 315
- 8.1.3 618 项目经理需要承担的角色 / 316

8.2 618 项目管理实践 / 316

- 8.2.1 启动阶段 / 316
- 8.2.2 规划阶段 / 317
- 8.2.3 执行阶段 / 319
- 8.2.4 监控阶段 / 325
- 8.2.5 收尾阶段 / 326

8.3 项目成果 / 326

参考文献 / 329

第1章

京东618 前世今生

宝剑锋从磨砺出。618既是全民年中购物节，又是购物的狂欢盛宴，更是京东研发的年中大考。考验也是机会，京东已经构建起纵贯电商交易的全链条，前台、中台、后台协同作战，支撑天量订单、万亿交易规模的强健技术体系。



1.1 京东618研发奋战之路

6月18日，这是一年365天中一个普普通通的日子，既不是中国传统节日，也不是重要纪念日，但它又是不平凡的一天。到了这一天，会有成千上万的人紧盯屏幕，面带笑容，看着自己心仪的商品，开始购物之旅！

让6月18日从普通的一天变成一个充满魔力的“全民年中购物节”，是京东坚守了十多年的正道成功的核心成果。

翻开京东的历史，618大促（以下简称618）的概念缘起于京东的店庆日，随着中国电子商务的发展经历了不同的历史阶段。由早期的京东红六月促销季到618京东购物Party，再到京东618品质狂欢节，最后发展为深入人心、备受消费者期待的年中购物盛宴。

相对于传统的商业业态，电子商务是基于线上服务进行的，自然需要现代IT技术的支持，618也不例外。

在人们狂热购物之后，海量交易产生的流量洪峰给京东系统带来巨大的冲击。为了保证良好的用户体验，由数千位京东研发人员日日夜夜坚守和一以贯之的偏执构筑了一道道坚实的防洪堤坝。

怎么让数以亿计的用户在这一天顺畅、快乐地享受购物乐趣，让用户不论使用PC、手机还是任何其他终端设备都能够顺利地购买到自己心仪的商品呢？这对于任何技术团队来说都是一个严峻的挑战。

1.1.1 发轫之始（1998—2003年）

1998年，京东集团创始人，董事局主席兼首席执行官刘强东先生在中关村创办京东公司，并租用了一个柜台，开始代理销售光磁产品，通过这种方式，他赚取了第一笔财富，这是京东所有事业的起点。当时的中关村是一个鱼龙混杂的电子产品集散地，但刘强东从一开始就和别的商家不一样，他坚持卖正品。

这一时期，也许刘强东本人也想不到，京东会以这么快的速度成长为中国最大的零售企业和中国收入规模最大的互联网企业，并成功进入世界500强。彼时的京东只是一个线下的小柜台，和互联网、电子商务没有任何联系。

不过这一阶段是京东发轫之始，是京东“正道成功”的价值观奠基之期。

2003年“非典”爆发，纯正店门可罗雀，大家都闭门不出，更别说出门购买光磁产品了。京东线下店的生意没法做，正在此时，为了生存，刘强东开始尝试在网上卖货，凭借在线下店积累的正品行货声誉，京东的电商之路开始了试水。

当时，京东的销售方式就是在一个专业论坛上发布自己售卖的光磁产品，如果有网友对其感兴趣，就通过论坛联系京东，再向京东汇款购买产品。

这一时期，严格意义上都不能说是电子商务，顶多可以称为线上销售。

技术支持？没有。

618？没听说过。

但是，这种粗陋的线上销售模式，让刘强东看到了网络的强大生命力，才有了后来的

京东商城公司。

1.1.2 奠基之初（2004—2009年）

2004年1月，京东多媒体网jdlaaser.com正式开通，淘宝、当当等已经逐渐步入正轨，而京东才刚刚起步。不过，这对京东来说是一个石破天惊的变化，它代表着京东正式向电子商务转型。

2004年，京东只有一组简单展示商品的网页，当时技术部门只是四五个人搭建的技术小组。京东的技术平台最初是外购的简单的ASP“个人网站系统”，是作为主要技术人员的刘强东，和他通过论坛销售光磁产品结识的曹鹏及另一位程序员共同开发的。在此阶段，京东的很多系统是这位社会学专业出身、自学编程的刘强东自己编写出来的。

当时的网站功能非常简单，用户可以线上下单，但是下单后却需要人工打印订单到仓库提货。这一阶段也有一些促销活动，都是限时特价，准备几个特价商品而已，618的概念也尚未出现。

2004—2006年，负责京东网站开发和管理人员不足5人，所有与系统相关的工作都是这几个人合力完成的。2005年上半年，刘强东力排众议关闭了线下的12家门店，把所有的资源都投入到了线上。随后，网站的销售额以每年200%的速度增长，到2007年6月，网站日订单量突破了3000单。

2007年8月，京东拿到了第一笔风险投资，企业发展进入了快车道，京东开始大力招募人才。2008年5月，曾经担任天极网CTO的李大学加入京东担任技术负责人。经过研究，他认为现有ASP架构已经难以支持业务发展，于是向.NET转型。

2008年，618正式推出。

当时只有3个技术人员备战618，实际工作就是值班。订单峰值也没有现在这么科学的评估方法。另外，京东还刚刚把网站从ASP架构升级到了.NET架构，压力非常大，最后好在整个网站没有宕机，经受住了考验。

当时所谓的618，远没有现在这么多丰富的内容，只是选择一些商品，设定特定的数量，定时进行秒杀，秒杀开始时间和结束时间都由人工控制，也没有什么反欺诈之类的系统存在。当时的618，不要说和智能相关，简直可以说和技术不相关，全靠“人肉”来处理。

2008年10月，京东上线家电类商品，开始向综合型电商转型。

和2008年相比，2009年的618多了一些促销活动，技术部门跟进做了一些设计、开发工作，不过总体上这个时代的618还尚在萌芽状态，处于相对粗糙的状态，技术部门还没有备战的概念。

1.1.3 收获之期（2010—2016年）

在2008年京东开始从ASP转向.NET架构之后，新版网站用户体验更好，每日订单量从几千单增长到十几万单。随后，京东又从.NET架构逐渐转向Java架构，在经过一段时间的.NET和Java架构并行之后，最终实现了基础Java的整体系统架构。京东也进入了一个高速发展阶段，618的影响力和订单量呈指数级增长。业务量的高速增长对于公司整体来说是



一件好事，但是对于技术人员来说却面临着巨大的挑战。

1. 618订单飞涨，痛并快乐着

2010年，京东成为国内首家销售额超过100亿元的网络零售企业。京东商城注册用户已超过6000万，遍及全国，入驻供应商近万家。在线销售家电、数码通信、计算机、家居百货、服装服饰、母婴、图书、食品等12大类，数万种品牌，百万种优质商品。

2010年3月，由从美国回国不久的京东技术顾问马松担纲，建设POP（Platform Open Plan）第三方商业平台，彻底改变京东之前全部为自营的经营模式。2010年12月开始全面运营，POP业务上线后，迅速为京东发展带来了新的驱动力。

随着京东的不断发展，人们对于京东正品行货的信任和京东高效物流的便捷，让618的流量屡创新高。尤其是瞬时流量峰值是惊人的，个别系统能达到平时的几十倍到几百倍。

在海量的数据及洪水般的流量面前，所有的问题都会被重新定义，任何简单的事情都会变得复杂。从电商的前端网站、订单、结算、支付、搜索、推荐，到后端的仓储、配送、客服、售后，以及采销人员使用的各种业务系统都面临前所未有的挑战。京东近7000个系统，几万个应用，任何一个出现问题，对销售活动的顺利进行，乃至对于京东的整体品牌建设都会有直接的影响。

当时，京东的系统还远远谈不上健壮，到了618的时候，研发人员有时候就变成了“消防员”。一方面，研发和业务之间存在有效沟通不足问题，不知道公司整体促销方案是什么样的，流量是从哪儿来的，又要引到哪儿去，哪些环节可能会成为流量集中突发的点。只能等着用户和采销人员反馈说哪里不行了，然后研发人员再扑上去看应该怎么处理。另一方面，当时基础服务很薄弱，没有监控系统可以实时查看系统的性能健康情况。即便系统出了问题，也没有方便快捷的方法来帮助诊断，因此当时能想到的最快的处理方式就是简单粗暴地重启系统服务。

2. 凡事预则立，618备战启动

正是由于存在各种各样的问题，从2013年开始，京东较为系统地利用项目管理的理念，并以项目形式推进618技术备战。

2013年的618备战，当时开会、备战都在北辰会议中心的一个叫作成都的会议室中进行，把该会议室作为备战的指挥部。这是京东第一次有了集中备战的概念，会议室当时只能坐40多人。2013年6月17日晚上，会议桌上放了几台台式机，墙边放了两台大电视，没有大屏。因为6月17号晚上在成都会议室挤了50多人，插座不够，就临时把机房的一些PDU拿过来用作插座，结果电量超负荷导致跳闸，最后只能从隔壁扯了一根电线接上插线板才勉强撑过了618的凌晨高峰期。

为了保障618的顺利进行，在技术层面，首先建立起完善的系统监控系统，其次把以前忽视的业务基础架构部分补足，然后通过充分的预案来解决出现的问题。

2014年3月，京东与腾讯开展了战略合作，QQ网购、拍拍的电商和物流并入京东。在这之后，京东逐渐形成了PC、App、微信、手Q四大入口，流量的快速加入加上京东日渐完善的物流体系，京东发展更为迅猛。

2014年4月22日, 京东在美国纳斯达克上市, 是中国首个在美国成功上市的大型综合类电子商务企业。

2014年9月, 京东搬到了亦庄的朝林广场, 环境好了很多, 也开始有了大屏展示, 备战条件更充分了。

伴随着时间的推移, 京东的618不仅仅在技术上有了充分的准备, 在组织上也有了横跨整个京东研发体系的项目备战团队, 都是由各个部门的精兵强将组成。

618在这一时期逐渐变成了一个家喻户晓的电商购物节。整个京东的IT基础体系也实现了从.NET向Java的迁移, 逐渐实现了服务化, 各个系统之间通过中间件进行衔接, 形成了完善的前、中、后台综合服务系统, 真正实现了对亿级天量流量电商网站的完美支撑。

在这一时期, 京东的交易额在2012年就突破600亿元, 到了2016年更是达到了创纪录的9392亿元。从2004年起, 12年间增长了9万倍, 平均年复合增长率超过150%, 技术在其中功不可没。

1.1.4 未来已来(2017年以后)

步入2017年, 618从技术+零售、品牌+零售、IP+零售、金融+零售、多终端+零售五个方面入手, 全面联动线上线下、引入消费者的参与和品牌商的协同。京东在618中的角色也由渠道商转变为中国零售基础设施提供商。

现在的618已经不仅仅是京东自己的618, 京东已经把618品牌向全社会开放, 开放618标志, 并无偿、不排他地授权包括竞争友商在内的所有企业使用。

2017年6月1日0点至2017年6月18日24点, 京东2017年的618整体活动累计下单金额达到1199亿元, 又上了一个新的台阶!

现在的618已经不是过去三五个人值班就能搞定的“小事情”, 而是彻彻底底成为了一个京东所有技术团队统一攻坚的核心“战役”。

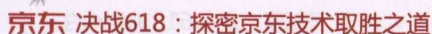
通过618技术备战, 不但支撑了集团的销售目标, 保证了系统安全平稳, 而且在备战过程中为人工智能、机器学习、大数据、弹性云等新技术提供了应用和验证的场景, 进一步夯实了系统基础。2017年更是首次实现了贯穿所有重要系统的全链路压测, 让研发人员在618之前有更充分的准备, 消除系统瓶颈。

在技术层面, 618备战期间, 各团队都要进行各种系统和软件的架构升级、代码优化、漏洞排查等性能提升工作, 还要有针对性地对系统、网络进行扩容。

要应对各种可能存在的对网络、对应用的攻击; 要做好系统监控, 提前根据业务预估, 进行流量压测; 梳理应急预案, 对预案进行实战演练。总之, 目的就是确保系统的高性能, 在遇到异常情况时都能快速处理, 并可以进行系统、机房及网络的各种切换。

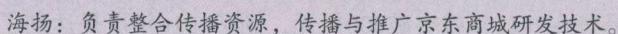
伴随着京东整体技术水平的不断进步, 备战的智能化日益提升, 很多预案都可以利用机器自动执行, 人在其中大都是起到查漏补缺、监控的作用。

在组织层面, 京东也已经形成了完善的备战体系架构。由备战总指挥统一进行部署, 618项目备战团队跟进, 统一协调整个京东横跨商城、物流、金融等多个部门的所有研发团队。通过备战启动会、周例会、日例会、誓师会等形成系列化的组织沟通机制。



刘强东强调，下一个10年到20年，零售业将迎来第四次零售革命。这场革命改变的并不是零售，而是零售的基础设施。零售的基础设施将变得极其可塑化、智能化和协同化，推动“无界零售”时代的到来，实现成本、效率、体验的升级。

未来已来，崭新的黄金时代正在来临！

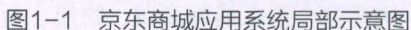


1.2 京东618技术备战概览

1.2.1 技术备战工作重点

1.应用系统

京东商城有着极其复杂的应用系统，如图1-1所示，包括PC、M站、App、微信手Q等多端融合的网关系统、搜索系统、广告系统、促销系统、订单系统、商家系统、供应链系统、支付系统、物流系统等。所有核心应用必须能够优雅地抵挡618零点峰值流量。因此，对各个应用系统研发团队而言，在保证系统高可用之后，618备战的主要工作是优化性能，“快”是不二法门。



2.基础架构

基础架构包括容器集群平台、商品图片系统、MySQL数据库、消息中间件、缓存中间件、RPC服务框架、流量接入与负载均衡、安全与风控等一系列技术服务。如图1-2所示，这一层所有系统的集合统称为JDOS——京东数据中心操作系统，统一调度多个数据中心的数万台服务器资源，提供共享技术服务来支撑所有应用系统，因此是每年技术备战的关键环节。基础架构的稳定性甚至直接决定618的成败，因此主要备战工作是通过容量扩展与故障演练来保障各种极端情况下服务的稳定性，“稳”是中心目标。

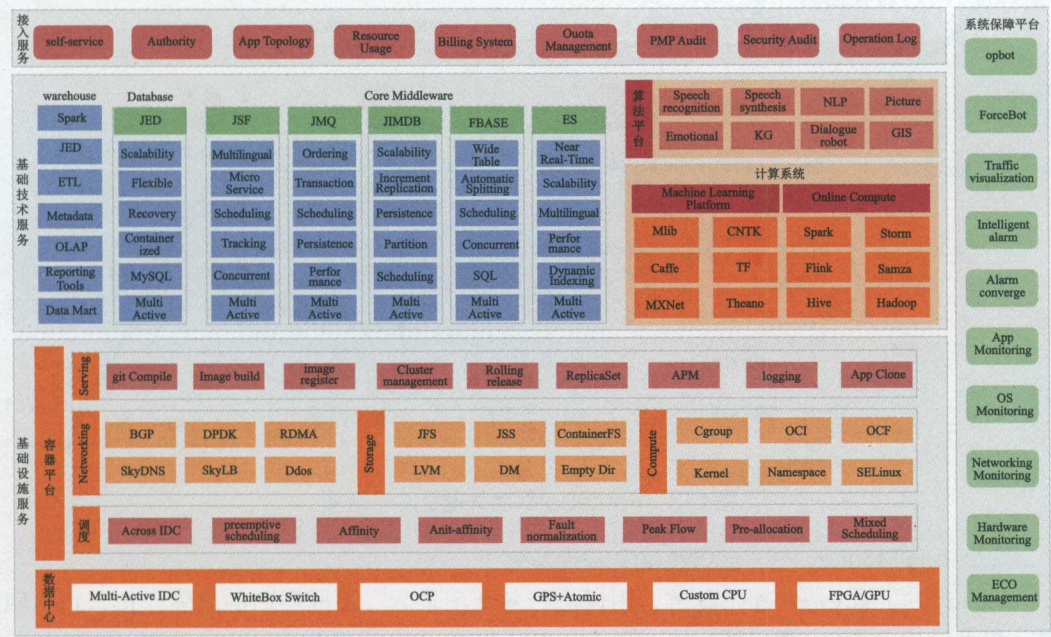


图1-2 Jingdong Datacenter OS（JDOS）

3.资源服务

资源服务包括数据中心网络、CDN、服务器等硬件资源。为应对618流量冲击，备战期间研发团队需要进行科学的容量规划，对服务器数量、IDC入口带宽、IDC间专线带宽、各地CDN节点进行扩容，为所有技术平台与应用系统提供充足的粮草弹药，“足”是唯一诉求。

1.2.2 技术备战利器

“工欲善其事，必先利其器。”数年的618备战，研发团队不断积累经验，设计并制造了两个重要武器——ForceBot与Chaos Monkey。

1.ForceBot——全链路军演压测系统

ForceBot意为“军演机器人”，它部署在京东各地的CDN节点，能够发起模拟真实用户的、超大规模的、购物流程全覆盖的访问流量，以便进行有效的军演压测，如图1-3

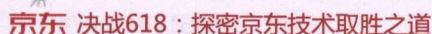


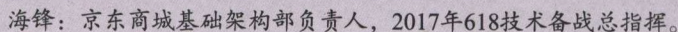
图1-3 ForceBot军演关键链路示意图

2. Chaos Monkey ——全场景故障演练系统

Chaos Monkey是一套工具链，用以进行各个场景的、不同层面的故障模拟，包括网络、机器、OS、DB、中间件、核心服务、流量入口、IDC。通过Chaos Monkey，各团队可以验证自身系统在出现故障情况下的可用性表现与对应的降级能力。以2017年618为例，研发团队共进行了70余次Chaos Monkey故障演练，找出薄弱环节近20个，做到了胸有成竹。

数年的618备战，京东总结了资源分配的“黄金法则”——9173原则。即90%的流量平稳型系统、10%的流量波动型系统，分别获得70%、30%的资源限额。所谓大道至简，这条朴素的经验，屡试不爽。

每年618购物节，也是京东技术团队的战场。追求完美，京东一直在路上。



第2章

万亿交易

丝滑平顺，坚如磐石

生意兴隆通四海。支撑用户在京东顺畅选购、极速下单、便捷支付的，是背后一整套经过历次618大考验证的黄金交易流程。不论618订单有多么凶猛，京东人都能挺住！





2.1 永不消失的网站

基于网站的重要性，马松总提出“永不消失的网站”，网站有任何一个细小的问题，都会影响用户的购物体验，为此我们从设计原则到高性能、高吞吐量、高可用的系统设计，到高灵敏的监控系统构思，再到应急方案的制订钻研了一套体系，本节将从网站首页、商品详情页、分类列表页、网站监控工具四个相关系统做详细讲解。

2.1.1 首页优化之路

京东首页是京东网站的门户，承载了所有重要的业务入口，流量很大，具有不言而喻的战略地位。

在浏览器里输入www.jd.com，进入京东首页，给用户呈现的是一个拥有各种商品和促销导购活动的缤纷页面，在这个页面的背后其实承载了一条艰辛的京东首页优化之路。

1. 整体架构

京东首页的整体架构示意如图2-1所示。



图2-1 首页架构图

从图2-1中可以看出，京东首页主要分为三层：前端展示层、中间逻辑层、底层数据层，其中前端展示层负责页面展示；中间逻辑容灾层则保证前端展示数据的实时性；而底层数据层作为最后一道关卡，负责整体数据的存储。

另外，CMS后台和容灾后台作为后台系统对上面三层在容灾方面做有力的补充，一起保障了首页的稳定。

下面就京东首页三层做具体的介绍。

2.前端展示层

随着业务发展，前端展示层主要经历了：静态、伪静态、动态三个阶段。表2-1对比了各个阶段的优缺点。

表 2-1 首页前端展示层的演变过程

阶段	优点	缺点
静态	静态页面维护简单，节省资源，四台 16U 的服务器配合 CDN 即可以支撑 618 峰值	首页各模块间存在强依赖，一个模块出现问题，会影响其他页面；同时在保证 CDN 的命中率方面，往往会设置比较长的 CDN 缓存时间，但已经无法跟上 618 业务量的提升步伐
伪静态	为首页各模块解耦，提高页面维护效率，结合 SSI 技术，在拼装静态页面的过程做动态业务逻辑，比如获取客户端 Cookie 展示简单个性化数据	个性化数据出现问题，无法满足千人千面的要求，势必做出改变
动态	随着 BI 的发展，个性化需求也越来越强烈，在保证首屏加载速度的同时实现千人千面，首屏同步渲染，保证首屏的加载速度；个性化数据应用层灾备，保证首页数据的完整性	易受第三方接口影响

3.中间逻辑层

中间逻辑层作为数据存储及数据传输的中间环节，是整个首页后台的核心，主要包括 CMS 数据维护运营系统和首页容灾系统两部分，与中间件一起保证了首页数据的完整性及实时性。

◆ **数据校验。**因为京东首页曝光率很高，依赖的服务也有很多，所以我们对每个依赖的服务进行了严格的校验，并检查数据内容是否符合规范，是否有空字段抛出。如遇异常情况则自动降级，且返回一份几分钟前的无错数据，保证数据的有效性，杜绝出现天窗情况。

◆ **多级缓存。**当 618 流量到来时，并发量剧增，某些后端服务压力巨大，会造成服务响应时间增加，从而会影响整个页面的数据加载速度，所以在既要保证效率，又要展示个性化数据的情况下，技术人员在设计中使用多级缓存来尽量减少后端服务压力。

◆ **降级开关。**为了应对 618，防止首页出现异常情况，我们设计了自动和手动的降级开关。同时充分建立各类系统容灾方案，保证首页在极端情况下可以随时切换至上一个正常版本，从而确保京东首页的正常展示和运行。

◆ **托底页面。**为确保京东的门户网站成为“永不消失的网站”，我们做了托底页面，其中包括 CDN 托底页面，定时向 CDN 推送一份包含所有请求的静态文件夹，这样保证在源服务器网络异常不能访问的极端情况下，还能有一份完整的页面展示给用户，从而确保



“京东首页永远不会消失”。

◆ **前端降级**。对于像京东首页这种大流量的网站，后端接口可能偶尔会出现错误，或者直接宕机，尤其是在618这种很容易达到流量峰值的时候，但是不能因为接口出错而使得页面显示出现错误，这就需要前端来配合给出一套合理的灾备方案。

这样的层层保障，很好地保证了页面的完整性。而且，针对所有接口，前端均有数据校验逻辑，每一个后端接口都要经过前端的数据校验，来验证接口的可用性，一旦接口数据异常，前端将主动调用托底接口来替代，这样来保证页面不至于错乱。

4. 底层数据层

这一层的主要功能是数据的存储备份和数据质量的提升两部分。

数据的存储备份：主要是接入公司大数据平台，做到数据的整体调度，包括运营数据、业务数据等，并进行分布式存储，互通互用。

数据质量的提升：主要是数据计算推荐，引入AI算法，通过人物画像分析用户行为，实现千人千面，提升转化率。

2.1.2 商品详情页的架构变迁

商品详情页是在电商交易中将商品直观、全面地呈现给用户，并对用户购买意愿产生很大影响的页面。为了保障用户在购买环节中流畅舒适地浏览体验，京东商品详情页随业务发展经历了三次大的架构版本调整，当商品数增长至数十亿级的时候，京东商品详情页架构的特点可以用一个关键词概括：异构+动态化，具体如图2-2所示。

下面将详细地介绍商品详情页架构的关键技术点、存在的主要问题及其解决方案。

关键技术点

◆ **基础中间件从MQ升级到JMQ（京东分布式消息中间件），从SAF到JSF（Jingdong Sewice Framework），大大提升稳定性。**

◆ **将数据与模板分离，利用Nginx+Lua技术，从前端JIMDB（Jingdong In Memory Data Bas）集群获取数据后做模板解析，生成完整的页面，最终输出给用户。**

◆ 多级缓存。

①**后端异构集群**。构造数据的应用接收到变更消息，按SKU、商品、商家、分类、品牌等维度拼装好数据后存入集群，该集群的维度粒度相对较细。

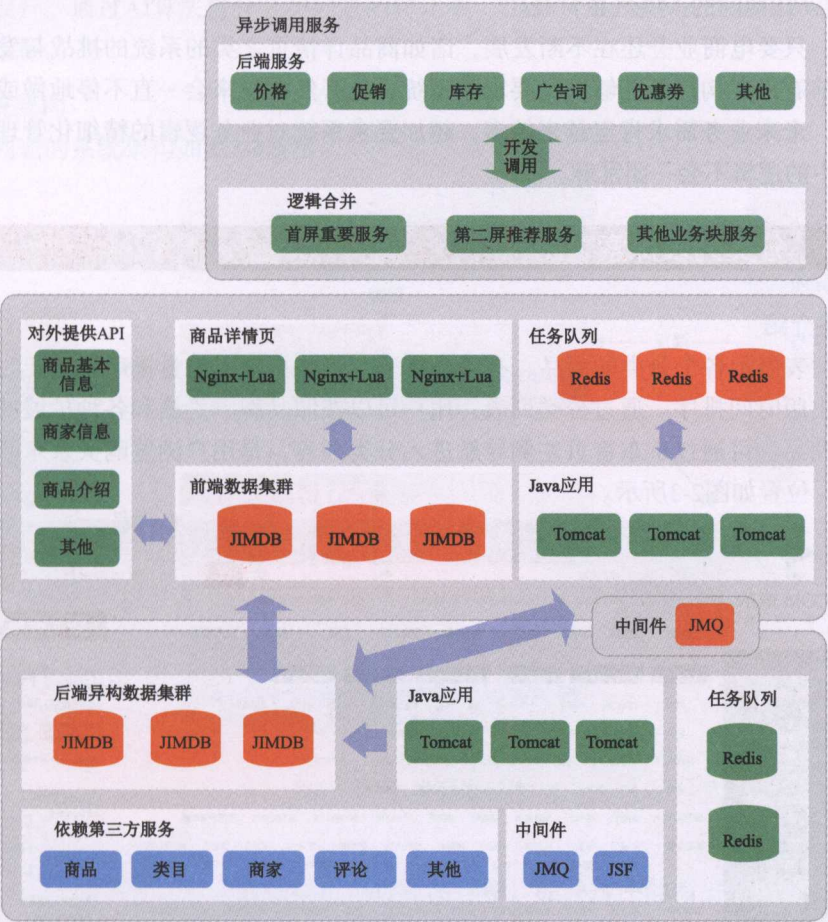
②**前端数据集群**。后端异构集群数据变更后，同样以消息通知，前端数据构造的应用将这些维度的数据进一步合并后存入集群，该集群的维度粒度相对较粗，由前端直接读取。

③**Nginx共享缓存**。这部分缓存能容纳的数据有限，一般只做分类、商家、品牌等这些总数据量很小的数据，减少跨服务器获取数据的成本。

◆ **商品详情页面异步调用的服务**，新增类似网关的应用统一管理调用（称作统一服务），将前端JS异步调用的多个服务合并为一个前端服务，在减少前端JS异步请求的次数改善浏览体验的同时，可以更方便地针对单个接口做异常降级处理。

◆ **数据及逻辑的复用共享**。电商系统中数据的粒度非常细化，比如，商品详情页选择

配送地址后，切换的库存状态及相关配送文案，后端是数十个系统应用接口的数据交互，因其中逻辑过于复杂而容易出错。所以将这部分逻辑合并成一个接口做在统一服务里，在其他系统中得到了充分的复用。



存在的问题

- ◆ 商品增长越来越快，存储数据增长速度越来越快。
- ◆ 业务需求的增长越来越迅速，导致业务需求实现过程中代码质量的下降，以及线上Bug增多的风险的提升。

解决办法

- ◆ **冷热数据的区分**。京东网站即使拥有数十亿甚至数百亿的商品，但是日均数亿的流量访问所覆盖商品的占比很小，所以可以建立维护用户访问的热点集群，控制集群数据量的大小。目前主要从两个维度做热点区分。

①按商品的状态维度，比如，上架销售商品的读取量肯定远远超过下架暂停销售的商品的读取量，下架商品过期时间远远短于商品上架时间。



②指定时间无访问的数据过期删除，新访问的商品通过异步任务重新刷入热点集群。

◆ **模块化的开发**。在业务需求的开发中引入“模块+版本”的概念，系统的逻辑及模块版本的切换引入到系统中进行管理维护，目前运行中的也仅仅是一小部分模块，比如商品详情页在618期间的一些气氛背景。

其实，只要电商业务还在不断发展，诸如商品详情页之类的系统的挑战与发展就不会停止，目前看来架构性能能够满足要求的系统，也必然在将来会一直不停地做或大或小的迭代变更，未来业务需求肯定越来越多，越加要求系统对业务逻辑的精细化管理和把控，确保系统中的逻辑不会一团乱麻。

2.1.3 分类列表探索之路

1. 系统介绍

分类列表展示各个分类的商品，有综合排序、价格排序、销量排序、上架时间排序，图书还有出版时间排序。通过分类列表，用户可以按照品牌、价格和各种扩展属性筛选出感兴趣的商品。可通过京东首页左侧导航进入分类列表，是用户浏览购买京东商品的主要入口之一，位置如图2-3所示。

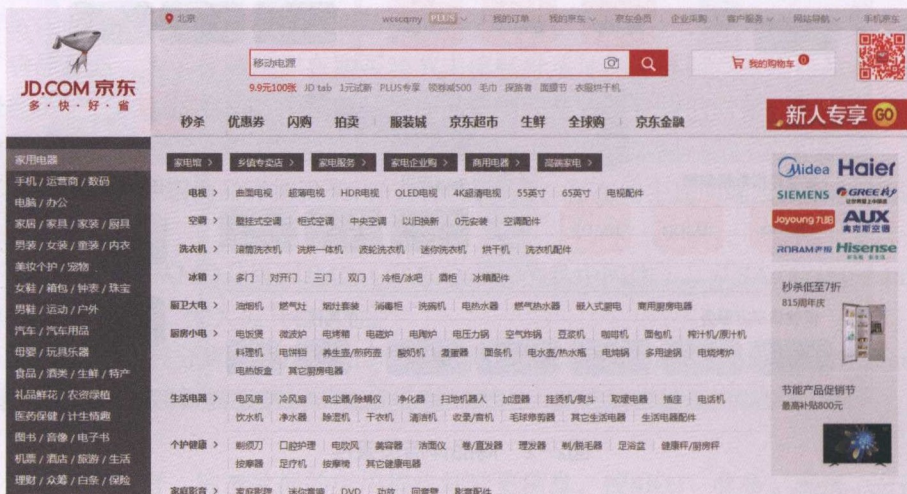


图2-3 分类列表入口

2. 系统特点

分类多。京东全站大概几千个分类。

商品多。每个分类商品多，有的分类多达上亿个SKU。

需求多样化。不同分类需求不一样，例如：大家电、生鲜、图书。

依赖多。依赖上游数十个系统，例如：商品、价格、库存、促销、配送等。

响应快。用户浏览页面的响应快，体验才好。

3. 架构设计目标

分布式。架构必须支持分布式，可以做到水平扩容。

高可用。做到多机房多活部署。

数据闭环。线上数据不依赖于外部API。

运维便捷。方便集群管理、分类配置等。

数据提升。通过AI算法提升分类列表的转化率、GMV（Gross Merchandise Volume）、客单价等指标。

4.系统架构

分类列表的系统架构如图2-4所示。

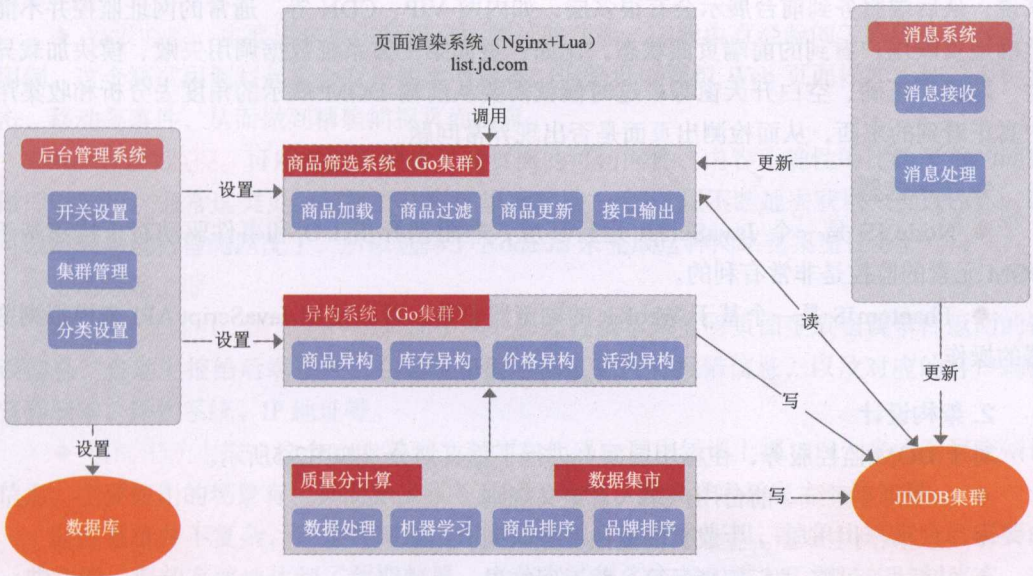


图2-4 分类列表架构

5.模块介绍

数据集市—质量分计算。数据集市负责商品排序、品牌排序的输出，其中影响排序的特征有30多个，例如：订单、评论、访问、搜索等数据。通过机器学习算法模型，计算出SKU、品牌的质量分，然后通过相应的质量分对商品和品牌进行排序，将商品排序、品牌排序推送JIMDB。

异构系统。异构系统负责将前台筛选、展示、排序的数据，从其他系统“异构”过来，保证系统数据的闭环。需要异构的信息包括：商品、价格、库存、配送、促销等。

商品筛选系统。商品筛选系统是采用Go语言开发的，这是为了保证筛选接口的快速响应而充分利用Go语言高并发的这一特性。商品筛选系统负责索引构建、索引更新和商品筛选查询。商品筛选系统分为线上和线下集群，线上集群提供商品筛选查询，线下集群负责索引的构建。线下集群索引构建完成，即可切换到线上使用。

页面渲染系统。页面渲染系统采用Nginx+Lua来实现，包含：模板渲染、防爬功能、缓存托底。

◆ **模板渲染。**通过筛选系统获取数据，渲染模板。Lua的模板渲染性能高。



◆ **防爬功能**。防止爬虫恶意刷量，保证系统的稳定性。

◆ **缓存托底**。将所有分类首页数据和热点数据缓存到lua_shared_dict中，在Nginx做Reload时，缓存数据不会丢失。

2.1.4 网站监控

通常在一个大型的 Web 项目中有很多监控，比如对后端的服务 API，接口存活、调用、延迟等监控，这些一般都用来监控后台接口数据层面的信息。但是对于大型网站系统来说，从后端服务到前台展示会有很多层，如内网 VIP、CDN 等。通常的网址监控并不能准确地反映用户看到的前端页面状态，比如：页面第三方系统数据调用失败、模块加载异常、数据不正确、空白开天窗等。这时候就需要从前端 DOM 展示的角度去分析和收集用户真正看到的东西，从而检测出页面是否出现异常问题。

1. 技术选型

◆ Node.JS 是一个 JavaScript 运行环境，非阻塞异步 I/O 和事件驱动对于构建基于 DOM 元素的监控是非常有利的。

◆ PhantomJS 是一个基于 WebKit 的浏览器引擎，可以使用 JavaScript API 来模拟浏览器的操作。

2. 架构设计

对于 DOM 监控服务，在应用层面上进行了垂直划分，如图2-5所示。

◆ **规则管理系统**。前台用户录入自定义规则。

◆ **规则队列生成器**。生成页面监控任务队列。

◆ **持续处理器**。进行规则运算，并返回结果。

◆ **PhantomJS 服务**。主要负责页面爬虫服务。

在应用层面上进行的垂直划分可以对应用进行分布式部署，提高处理能力。后期也方便做性能优化、系统改造扩展等。

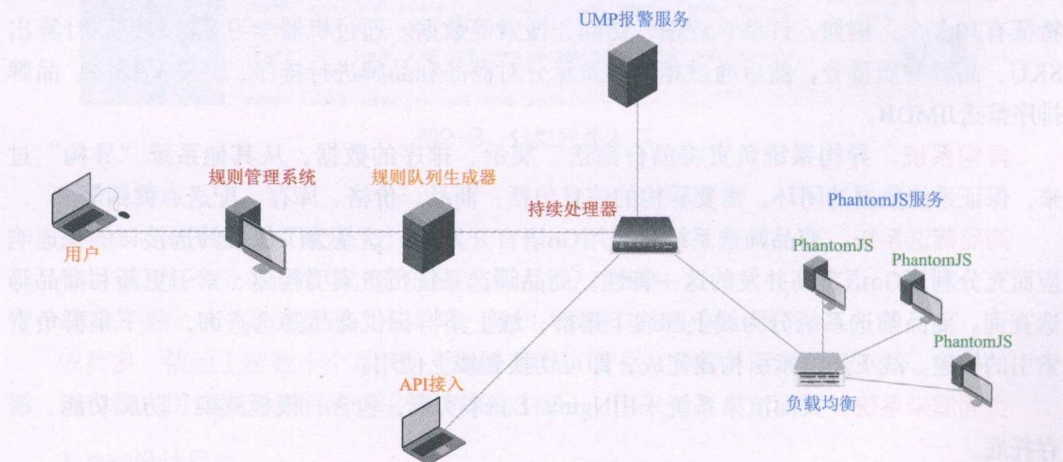


图2-5 网站监控系统结构

3. 解决方案

前台规则录入。这是一个独立的 Web 系统，系统主要用来收集用户录入的页面信息、页面对应的规则、展示错误信息。通过调用后端页面抓取服务来完成页面检测的任务，系统可以创建三种类型的检测页面：常规监控、高级监控、可用性监控。

◆ **常规监控。**录入一个页面地址和若干检测规则。注意这里的检测规则，研发人员把常用的一些检测点抽象成了一条类似测试用例的语句。每条规则用来匹配页面上的一个 DOM 元素，用 DOM 元素的属性来和预期做匹配，如果匹配失败，系统就会产生一条错误信息。

◆ **高级监控。**主要用来提供高级页面测试的功能，一般由有经验的工程师来撰写测试用例。这个测试用例写起来会有一些学习成本，但是可以模拟 Web 页面操作，如：鼠标单击、移动等事件，从而做到精确捕捉页面信息。

◆ **可用性监控。**可用性监控侧重于对页面的可访问性、内容正确性等比较严重的问题做即时监控。通常这类页面只需要在程序里面启动一个线程不断地去获取页面HTML，就可以对结果进行检测匹配了，所以选择了 Node.JS 来完成这种网络密集型任务。

主动错误上报

◆ **页面脚本执行错误监控。**页面引入一段监控脚本来监听页面生成错误事件返回的错误信息，自动上报给后端服务，在系统里面可以汇总所有报错信息，以及对应的客户端浏览器版本、操作系统、IP 地址等。

◆ **页面主动上报。**这个功能需要工程师在代码中调用错误上报 API，来主动提交错误信息。主要使用的场景有：页面异步服务延时无响应、模块降级托底主动通知等。

监控系统并不复杂，只是合理地运用了一些现有的技术框架，抽象出京东自己需要的一些功能，但却有效地达到了预期效果，也节省了很多之前需要人工测试的时间成本。并且在 618 战场上也发挥了巨大的作用，提前预警了一些 CDN 节点问题，及时报告线上页面图片加载错误、空白接口无数据等展示性问题。



xg：网站平台资深架构师，负责网站平台的架构设计开发。

2.2 交易平台

交易平台包含大家所熟知的购物车、结算下单、订单管理等核心功能，还涵盖了用户、商品、库存、价格、促销、优惠券等基础业务，同时还提供了诸多相关垂直与增值服务，如预售、抢购、评价、关注等，并且支撑自营与POP业务、实物与虚拟业务，为PC、移动端、微信、手Q、开普勒等前台业务提供共享服务。所有这些业务线完整贯穿了京东商城交易前、中、后三大环节，在不断发展中形成了如图2-6所示的业务架构体系。

因此，在整个电商体系中，交易平台无疑是最核心的一环，本节将从技术演进、架构揭秘、质量控制等方面分享交易平台服务是如何构建支持业务快速发展、支撑海量单量的



交易系统，以及如何保证其高性能、高可靠性。

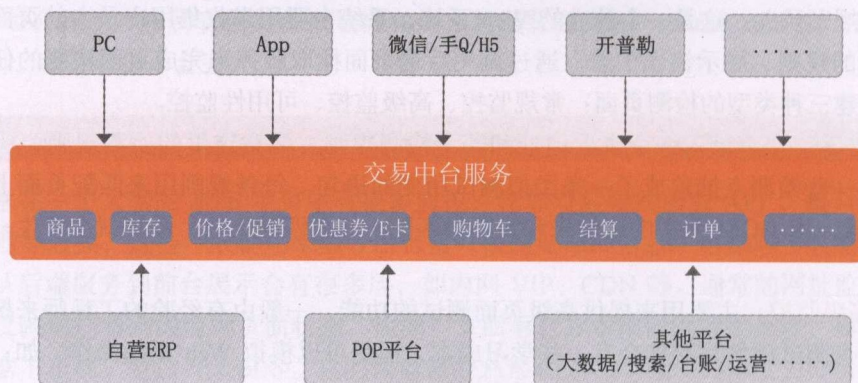


图2-6 交易平台的业务架构体系

2.2.1 交易平台技术演进

交易平台脱胎于京东商城前台网站，起初只是零散的几个小模块，并没有平台的概念。随着京东业务的高速增长，交易业务体系也在快速演化中，直到2013年，正式提出了交易平台化的概念。交易平台架构大体经历了五个阶段，如图2-7所示。

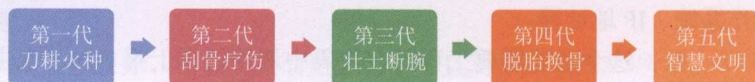


图2-7 交易平台架构的五个阶段

1.第一代（2004—2007年，刀耕火种）

最初版本的网站是从外部购买的，所有业务模块都在一个系统里面，数据库也在一起，架构极其简陋，连基本三层都没有划分，难以扩展和伸缩。开发语言使用相对落后的ASP，按现在流行术语就是单体应用。研发团队也很小，只有四五人，只能进行简单的修修补补。当然，这对发展期的京东来说，无疑是最快速、最适合的技术方案。然而，随着京东的迅猛发展，到2007年6月，日均单量已突破3000单，网站经常出现访问缓慢甚至无响应的情况，更经不起丝毫的促销引流活动，系统瓶颈已经非常明显。可以说，那时京东技术体系还处于非常原始的刀耕火种时期。

2.第二代（2008—2010年，刮骨疗伤）

2008年，京东迎来了第一任技术副总裁李大学先生。时不我待，系统升级迫在眉睫，李大学决定立即对网站进行重构。重构可以，但不能中断业务运营，不能影响用户购物，还必须要快。同时，考虑到团队和技术延续性，决定先过渡到C#.NET平台。历时三个月不分昼夜地快马加鞭，新版商城网站终于在2008年国庆节前夕成功上线。在这个阶段，产生了架构设计的理念，并开始应用主流的互联网架构。

经过一番改造，新系统逐步具备了一定的伸缩性，系统性能、用户体验也都得到大幅的提升，但仍然赶不上业务的发展速度。由于系统拆分还不够彻底，内部耦合度依然过

高，处理能力和存储容量还无法无限扩容，使得业务功能扩展困难，系统不具备足够的伸缩性，最终导致系统的稳定性并没有得到很大改善。而彼时，京东业务发展远超预估，单量已经从日均几千单增长到几万单，GMV达到200%的增幅，并且很快就超过了一开始规划的日均10万单容量。回头来看，这次重构并没能真正实现一套能够灵活应对高并发、大流量的交易平台架构体系，更像是一次“刮骨疗伤”。在这时期，技术还是疲于奔命地追着业务在跑。

3.第三代（2011—2014年，壮士断腕）

2010年年底，交易系统就提出要把架构提升到每天1000万单的容量，新架构还在规划中，便遇到了2011年图书5折大促。当时，促销效果非常好，但系统崩溃了。这次促销集中在3h内，用户提前将图书加入购物车，活动开始的一瞬间大量用户同时提交订单，库存扣减并发超过负荷，购物车和结算页频繁出现卡顿，甚至完全打不开，无法提交订单的现象，然而大量用户又不断刷新页面，恶性循环。当时，刘强东在微博上表示，要求增加3倍机器，第二天再搞一次促销活动。大家心里清楚，即使有30倍机器恐怕都难以撑住这样的高并发、大流量，因为系统先天不足，性能设计缺陷，没办法很好地水平扩容，架构“回炉再造”已经刻不容缓。现交易平台掌门人王晓钟那时刚好作为架构师加入京东，很快就接过了这个重任。

这次架构升级，主要有几个方面：开发语言全面转向Java，中间件全面转向开源产品，架构上全面服务化，大规模分布式，异步化，大量缓存乃至内存数据库的使用，最终形成如图2-8所示的架构。

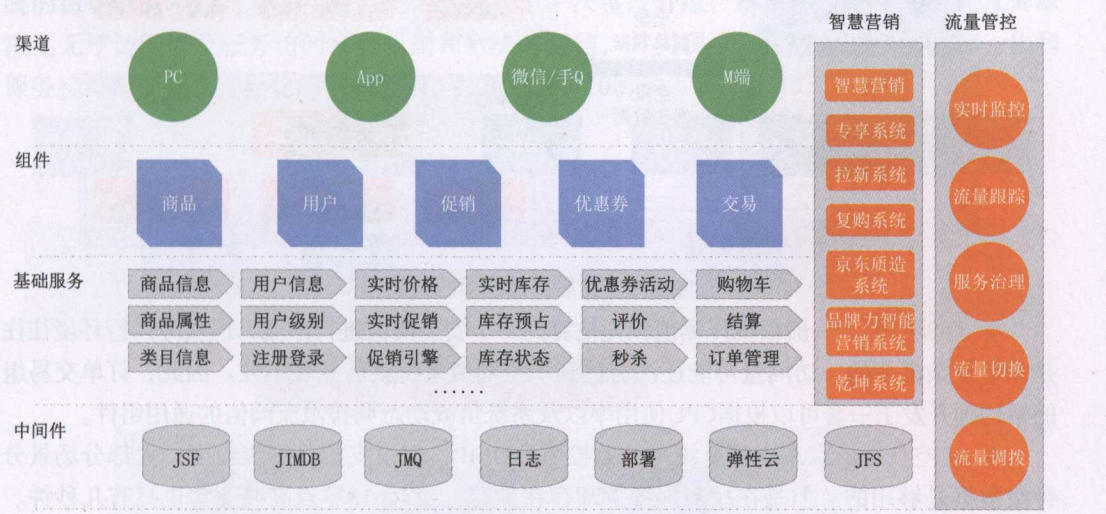


图2-8 交易平台的第三代架构

4.第四代（2015—2016年，脱胎换骨）

这一时期，交易平台主要在高性能、稳定性上深耕。

◆ **容器化**。2014年，基础平台开始预研Docker容器化，建立新一代弹性云。相对虚拟机，Docker有很多优势：更容易标准化，更少的资源消耗，更快速的部署。因此，



交易平台对新一代弹性云非常感兴趣。在基础平台的密切配合、大量性能和运维优化下，2015年618时交易平台已经有三分之一流量切换到Docker容器上，到2016年618时已经全部迁移到新弹性云上，极速扩容和快速部署的优势体现非常明显。

◆ **多中心**。2015年之前，交易平台已经将应用与存储都部署于多机房。应用是无状态的，所以很容易做到机房间的切换。但是存储是有状态的，当时机房间是基于主从同步实现的，主从同步是有延时的，当某一机房故障时，是不敢直接做切换的，可能会造成大量的数据不一致。因此，需要多个异地机房都能同时提供交易服务，任何一个机房出问题后都能由其他机房接管，用户尽可能在最近的机房完成交易成为新目标。这里关键的设计思路是通过用户路由分中心，使其在单中心内形成流量闭环，将数据同步回主中心，架构如图2-9所示。交易平台架构师设计开发了JingoBus以及RedisPipe，分别实现MySQL和Redis的跨双向机房同步。

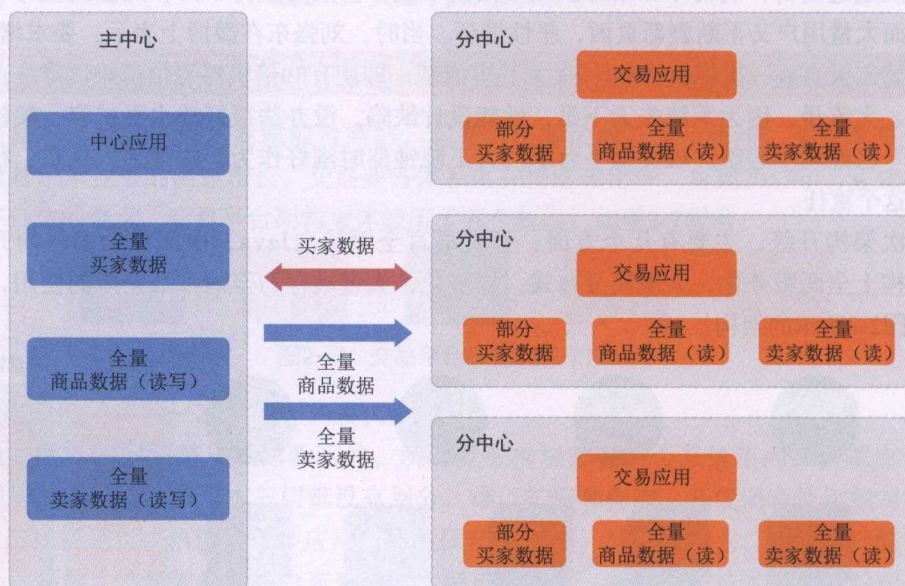


图2-9 交易平台的第四代架构

◆ **智能限流**。一般根据压测值预先估算的一个安全阈值进行限流，但实际运行环境往往是非常复杂多变的，访问量可能还没有达到安全阈值系统就已经扛不住。因此，订单交易组的架构师开发了一套可以根据CPU使用率以及系统负载动态调整限流阈值的通用组件。

◆ **监控升级**。京东内部有统一的监控平台UMP，但仅支持分钟级粒度，大部分场景分钟级数据是够用的。但是在故障报警上却存在延迟，而且618零点高峰通常也只有几秒钟。为了更快速地进行预警，更好地发现性能瓶颈，更准确地进行容量规划，交易平台迫切需要秒级监控数据。2016年上半年，交易平台运营组率先实现了核心业务指标的秒级监控，将用户登录量/注册量、实时订单量，按照PC、手机、微信等渠道以及省份、运营商、机房、品类、活动等多个维度进行细分，从而在及时发现核心业务指标变化的同时，能够快速定位、排查问题，并做出应急响应。2016年下半年，设计并实现了核心服务接口秒级监

控，这样便非常准确地监控日常峰值、大促峰值。

5.第五代（2017年，智慧时代）

正如刘强东在2017年京东开年大会上所说的，2017是京东的归零之年，也是京东技术新元年。交易平台将聚焦在大中台服务化战略上，提供可快速搭建各种形形色色新兴业务的组件，并利用大数据、人工智能在智慧保障、智慧交易上进行探索。

2.2.2 交易平台架构揭秘

随着京东商城流量的不断增长，交易平台的技术架构也在不断优化，从单体应用解耦到大规模分布式，从单机房部署到多中心交易。研发团队已经身经百战，备战不再手忙脚乱，而是稳如泰山。交易平台如今单接口日均最大调用量超百亿次，日均总调用量超千亿次，日均处理近千万订单，大促峰值单量是平常数十倍，而性能始终稳定在几十毫秒乃至几毫秒。本节将对交易核心系统最新技术架构进行重点阐述，揭示交易平台是如何应对如此巨大的流量与压力，如何在618中为用户提供流畅的购物体验的。

1.用户——最基础

在流畅自如的交易过程中，基础用户服务就像奠基石一样，它不露痕迹，却不可或缺，它像入海口一般，接纳来自四面八方的用户流量，迅速响应交易各个环节的海量访问。从2014年开始，京东先后与腾讯、沃尔玛进行深度合作，先后完成了与易迅、拍拍、微信、手Q、山姆会员店、一号店的账号融合互通，用户数呈指数级增长。为保障用户系统的稳定性，我们对基础用户系统架构进行了一次整改升级，解决了数据资源和配置资源混乱无序访问以及服务访问没有限制和章法的问题，并实现了用户数据资源调度统一化和服务接口精细化、差异化的治理机制，架构如图2-10所示。

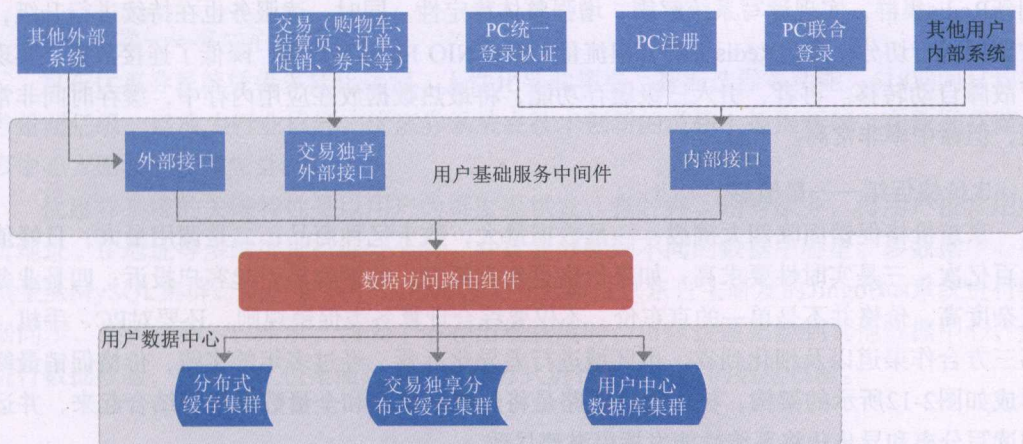


图2-10 架构升级后基础用户系统结构

用户基础服务中间件根据调用方的重要程度、调用量、访问的用户信息字段等对数据层的访问进行动态规划：主数据还是从数据，是否击穿到数据库等，针对核心交易调用的数据，隔离出独立的数据缓存，同时支持自动容错访问到共享的基础用户数据。在数据隔



离方面，按敏感度划分，敏感数据进行加密存储（如密码）、掩码传输（如昵称）；按访问频率划分，访问频繁的持久化存储到缓存，非频繁访问的进行临时存储；按调用方对用户服务的敏感级别划分，对敏感级别高的接入方，其用户数据单独隔离存储。在接口隔离方面，用户内部接口与外部调用接口隔离，数据更新接口和数据查询接口隔离，交易主流程依赖的服务接口跟非交易主流程依赖的服务接口隔离。这样就有效保障了交易主流程的可靠、稳定，经过了历年618考验，基础用户服务已稳如磐石。

2. 商品——最大量

正所谓，无商品不电商。如今，京东拥有海量的商品。如此海量的商品数据要准确传于整个电商体系的各节点，必须要有一套稳健、可靠的服务体系支撑。2015年年中，交易平台启动3.0架构升级，其理念是解耦、简化。架构越简单越好，简单即美，新人很快可以上手。架构也必须是松耦合的，写、读都可以各自升级，互不影响，最新架构如图2-11所示。

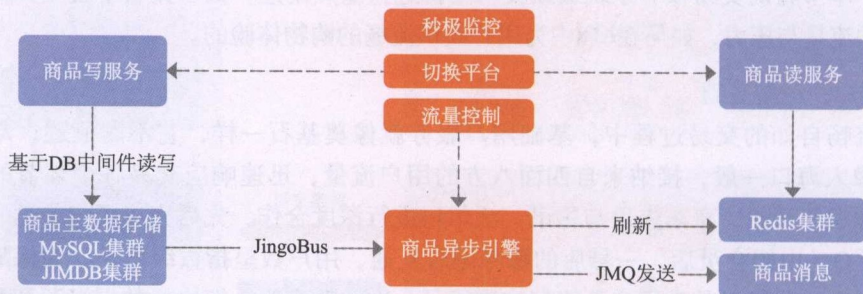


图2-11 商品系统架构V 3.0

首先，基于自研的JingoBus订阅商品MySQL Binlog日志更新，并发送商品变更通知、刷新Redis集群，实现读写系统解耦，增强整体稳定性。同时，读服务也在持续进行升级，按调用进行切分，多组Redis集群分担流量，升级NIO Redis客户端，降低了连接数量，实现了故障自动转移。再者，引入三级缓存功能，将最热数据放在应用内存中，缓存时间非常短，但命中率非常高。

3. 价格促销——最敏感

京东价格促销面临四大挑战：一是数据量大，数十亿种商品；二是调用量大，日峰值上百亿次；三是实时性要求高，如果价格更新有延迟，非常容易引起客户投诉；四是业务复杂度高，价格并不是单一的京东价，不仅要综合计算各类促销规则，还要对PC、手机、第三方合作渠道以及细化到省、市区域进行差异化运营。经过多年的实践，价格促销最终形成如图2-12所示的架构，核心设计思路是将热数据缓存和全量数据缓存结合起来，并运用读写分离和异步化将系统性能发挥得淋漓尽致。

价格促销系统核心技术主要是Nginx、Lua、Redis，有三个关键设计要点。

◆ **写逻辑：**后端调整价格或建立促销时，同步写入MySQL数据库，然后通过异步任务更新促销主Redis数据，同时更新价格主Redis的过期时间戳，通过Redis复制机制，将数据传播到各级从节点。这一过程，我们称之为“发布”。其中，促销Redis集群存储的

是全量数据，即作为内存数据库使用，而价格 Redis 集群则是基于被动过期缓存模式，同时如有变更，则主动更新以保证数据一致。注意这里有一个非常有意思的设计——价格 Redis 和 Nginx 是捆绑部署在同一服务器上的。这样，Nginx 通过 UNIX domain Socket 与 Redis 通信，避免了 TCP/IP 的网络开销；同时，每台服务器上的 Redis 集群是自治的，数据是闭环的，间接实现了无状态，可以快速扩容的要求。

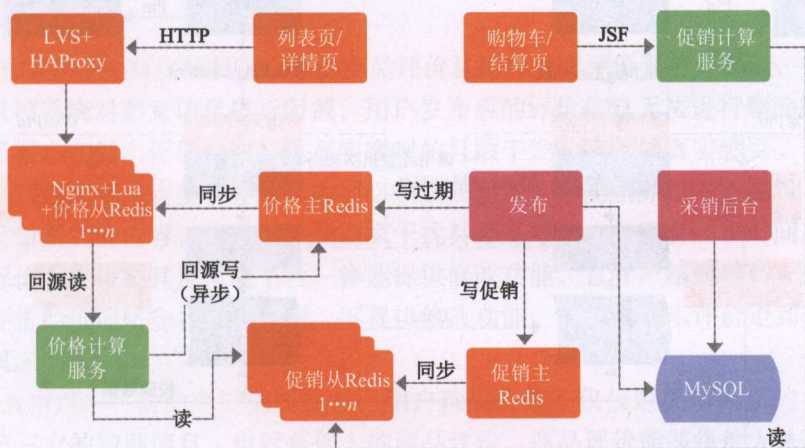


图2-12 价格促销架构

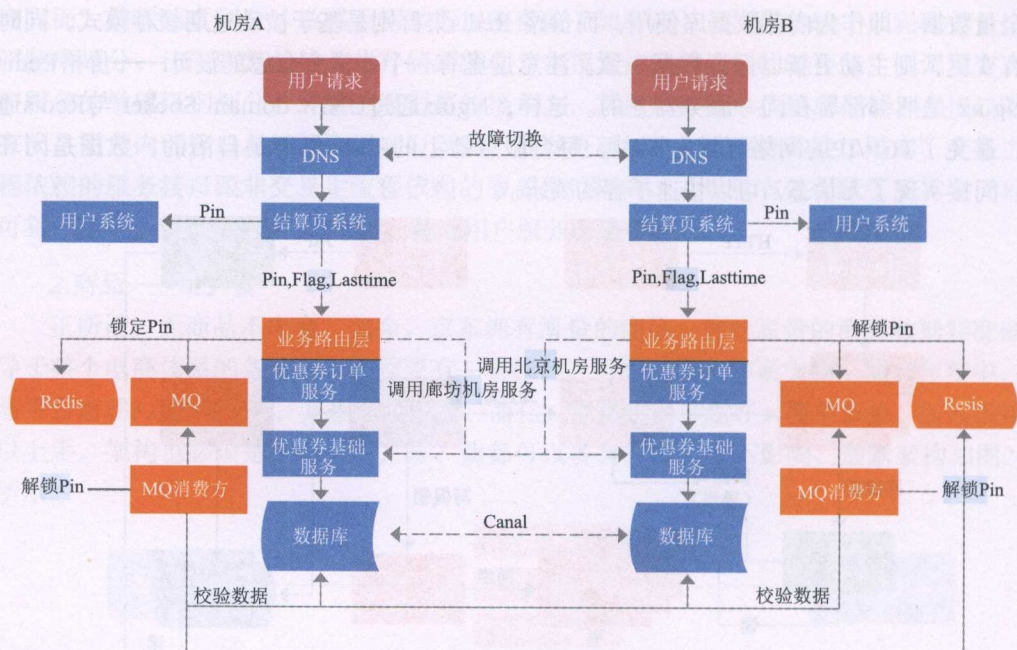
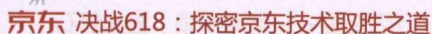
◆ **读逻辑**：访问价格实时服务时，此时内嵌 Nginx 的 Lua 程序直接读取本机 Redis（从节点）中的价格数据，若存在且无过期则直接返回用户；否则，将回源访问价格实时计算服务，然后返回最新价格给用户。

◆ **回源写**：价格实时计算服务调用促销计算服务，在返回最新价格给前置 Nginx 节点的同时，异步写价格主 Redis 集群，通过复制同步至各级从节点。

4. 优惠券——多中心先驱

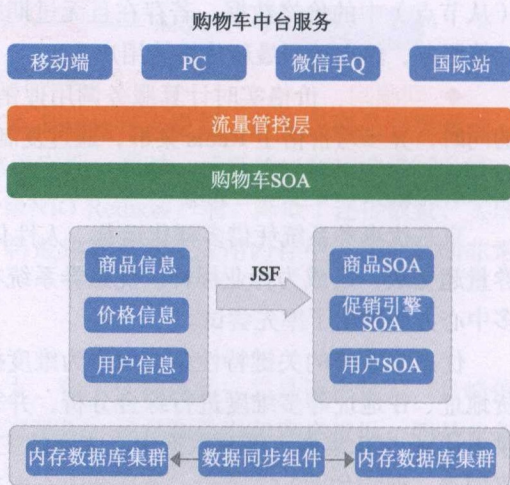
京东优惠券系统凭借多样化运营、人性化导购领券、智能选券等功能，618期间日均发券量超亿级，已成为行业标杆。优惠券系统在技术创新的道路上披荆斩棘，在数据存储、多中心方面做出了率先尝试。

优惠券系统的关键特性是以用户为维度提供发、查、存、用等服务，将用户按常用收货地址、IP地址等多维度进行综合分析，并智能分配到不同的数据中心里。多数据中心提供主从MySQL集群，同时数据中心之间的主节点通过京东自主研发的JingoBus系统进行数据同步，架构如图2-13所示。当某个中心出现故障时，可以快速切换到其他数据中心，并进行数据校验，校验方式也是通过单用户的方式进行最小粒度的数据校验。



5.购物车——剁手第一关

购物车中台服务聚合了用户、商品、价格、促销等服务，为PC、移动、微信手Q各端提供统一服务，也为后续结算下单提供了高性能服务。购物车作为用户交易流程的首道大门，618期间承担巨大的流量冲击，核心接口QPS在最近的几次大促中均达到百万级别，面对如此巨大的洪峰访问量，购物车系统选择了走接口轻量化、存储内存化、终端隔离化的路线。历经多年数次大促洗礼，购物车应用从单一机房部署到多机房部署，从单一分组到PC、移动多组隔离，而存储则从MySQL数据库到主从式Redis，再到多中心式内存数据库的转变，目前购物车中台服务整体架构如图2-14所示。



2.2.3 商品评价系统

1. 评价——给你客观的选品参考

商品评价是用户购物后的内容沉淀，也是用户内心真实的吐露，不仅从更多层面给予

其他购买用户提供相关信息，还促进着商城业务的快速发展，已经成为商城内容呈现的一个重要方式。

价值所在。商品评价是必须在京东购物以后，才能对购买的商品进行评价，用户只有购买后才对商品具有发言权，输出的内容最可信，也更具有参考价值。商品评价是内容的产物，其本质是影响用户购买、传递商品及服务多层次的数据信息，因此，它的价值从以下几个方面得以体现。

◆ **价值坚守——真实客观。**京东的商品评价从用户发起评价到前台展示，整个过程通过严格的风控系统对恶意信息进行拦截，用户发布后的评价信息无法进行删除操作，这就有效保障了客户在做出评价的时候是真实客观的且敢于发表实际的真实感受。自从有商品评价开始，刘强东就一直要求评价的内容必须真实客观，不能删除也不能编辑，为的就是保证客户反馈真实的内容，而且能不受商家干扰甚至是威胁。曾经有一段时间，商家要求京东商品评价系统也如其他电商平台一样能提供修改功能，且呼声越来越高。最终刘强东重申京东评价的价值坚守：真实客观，不提供修改功能。这也是京东评价走到现在始终坚持的价值观。

◆ **价值原点——帮助客户购买决策。**用户购买决策不仅仅取决于商品的主要信息，更取决于第三方的辅助信息，也就是他人的商品评价。商品评价能够将他人从不同维度和观点的使用情况反馈出来，对于潜在消费者具有非常大帮助，而购买过程中还可以通过咨询、回复等方式与评价者进行互动，亦可与商城客服进行互动，这对于改善客户关系是非常有帮助的。如果消费者在购物过程中或者购物结束以后和商城发生了除购物以外的更多互动，那么用户花费在与商城互动上的时间越多，对商城的信赖和忠诚度就越高，而商城互动内容就会更加丰富精彩。

◆ **价值传递——口碑积累。**随着用户群体的不断扩大、商品销量的步步攀升，积累下来的用户评价将是商城潜在的巨大财富，评价数多、咨询量大、好评率高一方面能够为商城积累更多的口碑，另一方面又提升了品牌的良好形象。用户购物时一开始是十分关注商品的属性的，但随着时间的发展，良好的品牌沉淀将是影响用户购物不可或缺的一部分，这一优势也在电商行业激烈的竞争中得以凸显出来。

◆ **价值追求——提升转化与流量。**用户购物下单是对自身需求的实现，而购买过程是一个可快可慢的对比过程，一件商品是否满足自己要求，通过商品介绍、商品描述便可知晓，而商品评价恰巧是这个需求实现的催化剂，通过文字、图片、视频等方式不断增强用户的购买信心，最终促成用户下单，在这个过程中商品评价起到促进和影响的作用，提升了购买转化。

原创内容、丰富的内容呈现形式，以及互动式的内容更新，搜索引擎甚是喜欢，提高了网站在搜索引擎的权重，也给商城引入更多的流量。

◆ **价值升华——倾听用户心声。**商品评价作为用户内容的载体，是用户真实心声的体现，不管是对商品质量，还是安装服务，抑或是配送服务，哪怕是买贵了等各种情绪，客户都会在评论里体现出来。2016年京东的智慧营销平台之一“倾听”平台的第一版本开始就从分析用户评价数据衍生而来，所以系统平台的名称就取为倾听，意为倾听用户的一切



声音，不断提升商品和服务的品质。

发展历程。商品评价的数据积累和价值体现离不开日渐增加的业务及系统发展，而商品评价系统从只有评分、标题、心得等简单的评价内容到增加了晒单、问答、标签、分享等功能，再到评价系统拥有每日流量数千万、上亿的数据存储，系统经过了一次又一次的蜕变，开始走SOA服务化路线，同时利用前沿的缓存技术解决流量压力问题，基础架构得到了夯实，系统稳定得到了极大的提高。

到了移动互联时代，为给予用户更丰富的使用体验，商品评价系统结合移动社交的优势，已拓展出用户扫码晒图、视频晒单、快速问答，以及语音问答等多种形式的衍生产品。在系统架构方面，评价文本存储由MySQL迁移到了MongoDB、HBase双存储上，有效地解决了大文本、大数据量的存储问题。而评价MySQL数据库通过交易系统的数据多中心实现了异地多活、高可用的架构部署，有力地保障了数十亿评价的数据存储。在服务治理方面，逐渐将不同业务、不同单元的服务进行微服务化，实现业务隔离、服务隔离、灵活管控的治理方案。同时接入秒级监控系统和故障自动切换机制，对系统健康状况进行实时监控，及时发现故障并自动进行故障修复，实现系统及服务的高可用。目前评价的架构如图2-15所示。

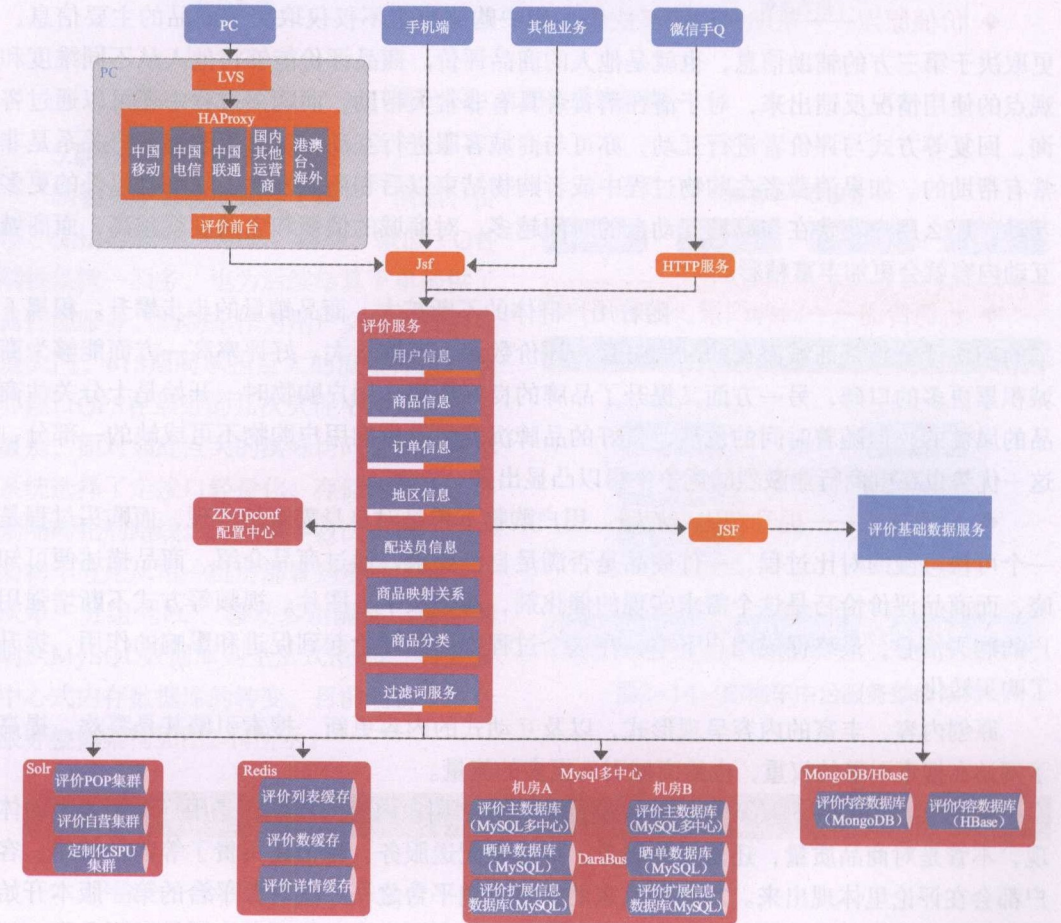


图2-15 评价系统架构

随着大数据、人工智能时代的到来，机器学习更是技术发展的趋势，评价数据经过多年的沉淀、积累，已经拥有足够丰富的样本数据，通过自然语言处理技术可以挖掘出评价的更多价值，同时还能帮助我们理解用户文字背后的深层含义，具体可从以下几方面得以体现：

①关键词提取。从评价内容中提取核心关键词，再将关键词生成标签，用户可以根据标签对评价内容进行筛选，通过精准的内容筛选来帮助用户进行购买。同时我们还可以通过标签进行评价数据统计与评价内容分析，来帮助我们改进产品质量与提升服务内容。

②内容分类。通过预设分类对评价内容进行文本归类，根据不同品类、不同品牌结合推荐算法向用户进行精准内容推送。

③情感分析。对评价内容进行精准分析，通过用户的文本内容解答出用户的情感倾向。

④黄反识别。精准地过滤评价内容中的黄色信息、政治敏感及反动内容。

⑤垃圾内容识别。精准识别出广告内容、不文明用语及低质量文本。

将这些评价内容“取之于民，用之于民”，给予用户更丰富、更精彩的购物体验，让用户的购物旅程变得简单、快乐！

2.问答——随心所欲，无所不答

用户购买后产生的商品评价，很大程度上帮助了其他用户进行购买决策，但用户决策过程中，如果想要了解更多的产品信息，以及评价以外更多的购物感受，仅评价功能是无法满足用户需求的。为此，问答系统诞生，旨在为用户答疑解惑，提升产品转化。

起初，京东的问答起始于商品咨询，用户将自己的问题以咨询形式提交，提交类型包括商品服务、库存配送、支付、发票保修、促销等几大类，客服人员根据不同分类进行回复，如图2-16所示。

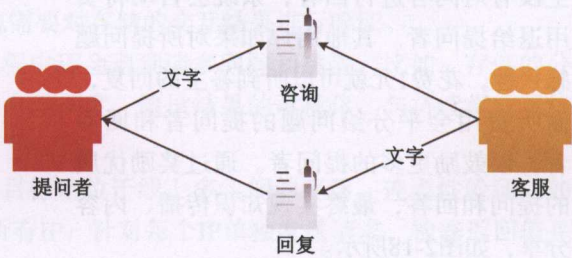


图2-16 问答

京东客服不仅为用户答疑解惑，还是用户很好的倾听者，内容也极其丰富。京东的客服MM的幽默与诙谐，时不时地也让用户心中泛起一片片涟漪。

后来，商品问答从咨询中脱离出来，成为升级版购买咨询，逐渐开放了回复方式，如图2-17所示，用户提交商品提问后，不仅官方客服可以进行回复，所有购买过此商品的用户都能够对问题进行回答。系统还会通过消息推送方式邀请优质用户参与回

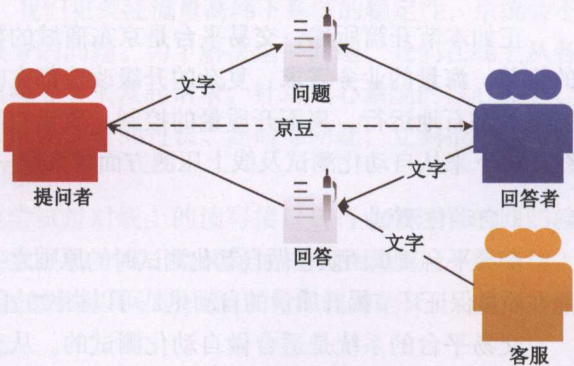


图2-17 升级版购买咨询



答，问题得到了及时反馈，提问者还能获取更多优质的回答。

最后提问者还可以选取其中一个作为最佳答案，最佳回答者将得到京豆奖励，问答系统同时还提供了京豆打赏功能，受打赏用户还可以得到额外的奖励，通过这些刺激，用户在贡献高价值内容的同时也得到了相应的回馈。

3.京答——传播知识、快乐分享

商品评价、商品问答为用户解决了不少购买决策上的问题，而在知识内容领域，2017年2月22日“京答”正式上线，成为付费短语音产品里的新生力量，帮助用户传播知识、分享知识。

“京答”与行业中语音付费产品“分答”形式上类似，但“京答”以图书为载体，更专注于图书音像领域，整合了各行各业的资源和作者，形成了丰富的IP资源，旨在给予作者更多推广平台进行内容推荐、知识传播，同时还帮助用户解决了读者与作者互动交流、文章理解、作品咨询等痛点问题。

京答系统交易环节由提问者、答主、偷听者三种角色组成，答主为自己的提问设定价格，提问者向答主先支付提问金额后通过文字形式进行提问，答主选择有价值的问题进行短语音回答，回答后交易完成，答主收到提问者的付款，如果72h内容答主没有对问答进行回答，系统会自动将费用退给提问者。其他用户如果对所提问题感兴趣，花费1元就可以听到答主的回复，偷听费用会平分给问题的提问者和回答者，以鼓励更多的提问者。通过奖励优质的提问和回答，最终实现知识传播、内容分享，如图2-18所示。

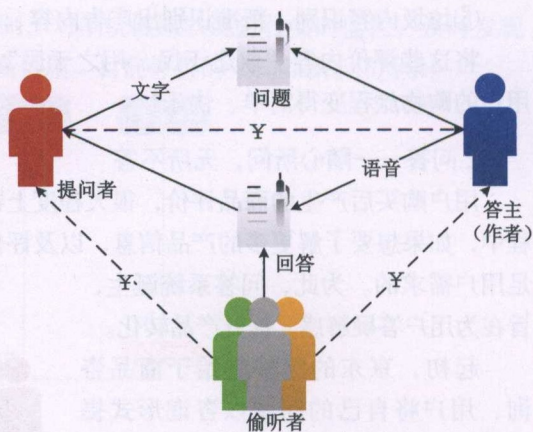


图2-18 京答系统

2.2.4 质量控制

正如本节开篇所述，交易平台是京东商城的核心平台，负责的系统支撑着日均上千万的订单，海量的业务需求、复杂的升级改造都在时时刻刻地考验着系统的稳定，要保障系统坚如磐石地运行，离不开质量的控制，质量控制是系统运行的保障，是交易发展的生命线。接下来从自动化测试及线上压测方面带大家一窥究竟。

1.自动化测试

交易平台质量团队在做自动化测试时的原则之一就是：能在软件开发流程中提升效率的、能在质量保证环节提升质量的自动化是可以去做的自动化测试，不为了自动化而去做自动化。

交易平台的系统是适合做自动化测试的。从交易系统的特点来看，交易系统多为后端服务，表现形势多是JSF接口、HTTP接口，所以我们的自动化测试也是基于JSF、HTTP接

口的自动化测试。跟UI相比而言，接口本身只有业务逻辑与底层存储的交互，天然上就规避了页面UI频繁变动的风险，也在一定程度上降低了由于UI变动造成的自动化测试用例维护成本高的风险，所以接口更适合做自动化测试。我们做自动化测试主要有两个途径：一是在测试环境上对核心系统进行回归测试，大幅减少日常回归测试的重复执行的工作量；二是在线上校验核心系统的接口存活，在上线后对线上的多实例（多则上千个实例）进行逐一验证。这两种方式虽简单，却有效，直指测试的本质：保证质量+提升效率。

在测试环境里，我们根据对手工测试的模拟，抽象出接口测试的统一模型，以适用于所有的接口，不论何种形态，不论什么协议，这个模型统统可以囊括。每一个接口测试用例都包含以下几个步骤：

第一、初始化测试数据。这个数据的初始化可以用不同的方式来实现，有的团队倾向于在跑所有的测试用例之前，对整个存储实例（Redis、MySQL等）进行重写，将自己在测试过程中需要的所有数据写进去；有的团队倾向于在跑每一条用例之前在存储实例中初始化本条用例依赖的测试数据；有的团队也会通过调用外部接口的方式来达到初始化数据的目的。不同的方式各有优缺点，在不同的应用系统中需要综合考虑。

第二、调用被测应用接口。对于不同类型的接口可能有不同的实现，对于HTTP接口，可以用Java的HttpClient实现，也可以用Python的Request模块来实现；对于JSF接口，一般可以模拟一个JSF的客户端，可以结合Spring来实现，也可以采用JSF支持的API方式调用和Generic方式的泛化调用。

第三、结果对比。可以对返回结果的整体、部分进行对比，如果接口本身是写接口，与存储（Redis、MySQL）有交互，那么也需要对存储的交互结果进行验证。

总体上分这三步，但是在实际执行过程中还会遇到诸多问题和挑战，比如：存储的分库分表、接口入参不能重用、测试执行的自动调度、测试结果的可视化、与持续集成系统的接入等，介于篇幅这里就不展开了。

线上环境的日常测试中，我们的初期目标定位于线上多实例的存活、连通性验证。如果是HTTP接口，可以遍历某个应用下的所有IP，针对每个IP单独发送请求，检查返回值是否符合预期。如果是JSF接口，我们可以针对每个分组，遍历分组下所有的实例IP，在上线过程中灰度发布，发布完成后进行验证，验证没问题后再进行下一个分组的发布。

在618期间的线上自动化测试场景中，我们更关注流量高峰下系统的稳定性，系统会不会出现因为性能问题造成响应慢、拒绝服务的问题。为了解决这一问题，我们在线上从各个流量入口（PC端、微信端、App端、M端等）来发起请求，针对核心系统的核心业务，在UI层面进行自动化巡检，一旦发现如页面响应时间过长、超时等问题，立刻报警，团队成员立刻介入，第一时间定位问题、解决问题。

在线上环境的自动化测试中，我们也尝试过对线上的读写接口进行业务逻辑的回归验证，只不过如何保证测试数据与线上真实数据的分离，如何保证线上测试执行不影响线上流量，如何对测试过程写入的脏数据进行过滤删除等，都是需要考虑的问题。

2.线上压测

交易平台质量团队自成立以来，非常重视用户体验，不仅对功能要求非常严格，对



性能的要求也越来越高。从最初的系统框架性能测试、服务性能测试、线下性能测试，到现在的线上性能测试，交易平台性能测试在不断深入、拓宽、探索、创新。以下主要从性能测试模型、性能测试流程、性能测试方法、性能分析四个方面进行介绍，如图2-19所示。

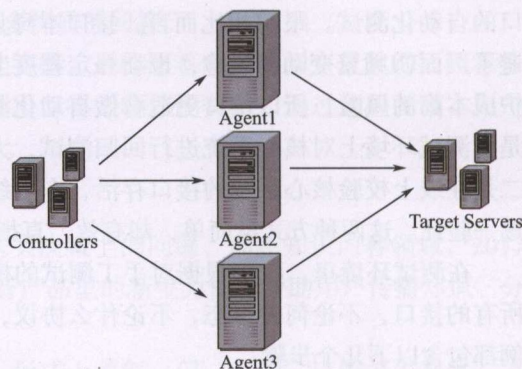


图2-19 性能测试模型

首先介绍性能测试模型。为了便于理解自主开发的压测平台，我们在实践中摸索出一个适用于当前现状的性能测试模型。模型由三部分组成：Web管理工具集群、Agent代理集群和目标服务器集群。在该模型中，Controller可通过HTTP协议调用Agent测试代码中的测试场景，再由Agent对目标服务器发起请求，以完成整个压测过程。该模型有三个优点：

- ◆ 管理端是Web管理页面，可以统一发送请求、统一收集数据，结果准确、可靠。
- ◆ 支持在线编辑脚本、启动场景，同时支持多用户同时操作，大大降低性能测试场景执行和推广的难度。
- ◆ Agent端支持水平扩展，通过J-one.jd.com（统一工作平台）任意扩展Agent，最大限度模拟生产环境的流量。目标服务器作为压力承受端，也可以水平扩展，模拟多服务器负载均衡、容灾等场景。

其次介绍性能测试方法。性能测试通常采用先单场景、后混合场景的执行方法。单场景是针对单个方法或单个页面，构建一个性能测试场景而进行的性能测试。

例如抽奖活动：针对抽奖方法特点，制定测试策略，创建性能、负载、压力、稳定性测试的场景。在618备战前，根据UMP历史监控数据，压测618调用量的10倍、20倍，甚至30倍，提前发现暴露的性能问题或者隐藏的性能隐患，有效地控制风险。为了尽量和生产环境运行的业务压力以及用户使用场景一致，测试系统的整体性能是否满足需求，通过UMP监控数据分析，按照比例模拟用户请求、并发，组合成一个混合场景。

3.性能分析

在分布式框架下，针对不同的系统应用、不同的测试目标，以及不同的性能关注点，根据性能指标的表现，采取逐步定位、从外到内、从表及里、逐层分解的分析原则。通常可按以下顺序进行：中间件配置（Apache/JBoss/JSF参数配置、数据库参数配置）→应用系统日志→应用系统瓶颈（线程池配置、代码、SQL语句、算法等）→第三方服务提供者性能瓶颈。（注：以上是通常采用的分析过程，在具体分析过程中，还需根据问题具体分析）

以上带大家领略了交易平台质量控制的一些基本流程和实践。可以看到，我们通过技术方面的不断总结和探索，让质量控制越来越强，也让交易系统越来越稳定。未来交易平台的质量将会结合部门战略，继续开发质量手段和提升测试技术。

回顾过去十多年，可以看到交易平台在早期是被业务发展推着走，可以说是业务成就

了技术，反过来技术又服务于业务，进一步促进了业务更加快速发展；在未来，交易平台将以技术驱动业务为战略，以中台化为目标，以智能化为手段，将现有基础服务抽象整合成为共享组件，将现有服务资源以多元化的方式展示给外界，通过灵活的配置、组织、编排，实现新业务域系统的快速搭建，以支持多形态业务需求。交易平台将致力于技术改变生活，在自动化服务、智能保障、智慧交易上努力拓荒，为京东第四次零售革命奠定坚实的技术基础。



王灵：交易平台技术专家，负责交易平台基础架构规划。

2.3 京东支付——更便捷，更省钱

伴随“京东摩尔定律”——每年的618订单量至少翻番，在0点时海量订单洪峰般涌来后，直接冲击支付体系的各个核心系统，直到支付机构和银行。支付体系不仅要保证快速完成支付流量分发、高效地完成支付扣款以及用户支付体验足够流畅，还要保证在面对支付机构、银行及流量过载导致出现突发情况时能够快速地完成自动降级和手动切换。

随着京东体量的快速增长，支付体系也在快速完成自我迭代。支付技术体系完成所有黄金链路架构的升级和重构，支付产品体系快速完成账户支付体系构建、支付功能产品的完善以及支付精准营销和智能运营产品体系的建设。

支付总体架构的升级重构实质是思想的转变和处理方式的革新。最初的支付总体架构是以主从数据库结构为核心存储、高耦合的应用系统以及集中式的部署方式。随着用户量、流量的增长，引发网络资源、存储资源、计算资源、I/O资源等成为阻碍系统高效流转的瓶颈，架构也随之逐步转变为对存储、应用和部署方式进行时间和空间上不同角度的拆分，如图2-20所示。

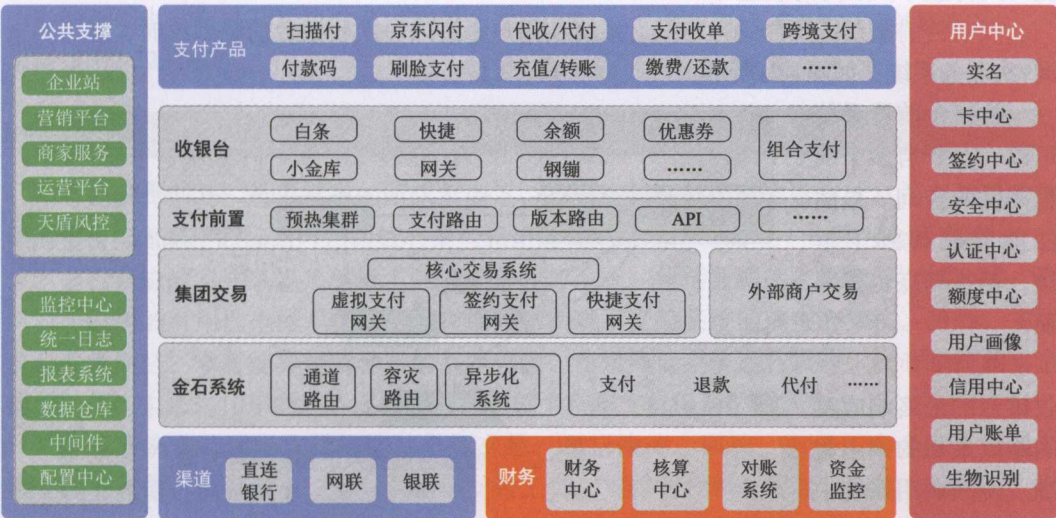


图2-20 支付系统整体架构



2.3.1 移动支付

1. 打造新移动支付

新版京东支付在2015年12月份的时候开始了系统的建设。首先，将银行卡的快捷支付功能做成原生页面，封装成SDK。而对外输出移动端原生页面，率先集成到京东到家App，后续推广到京东金融App和京东App。然后，支持了签约支付，用户可以设置支付密码，并使用支付密码完成付款；扩充了支付方式，支持虚拟资产支付，如白条、小金库、钱包余额、钢镚；后续版本又增加了组合支付，支持虚拟资产余额不足时可以使用两种支付方式组合完成付款。

2016年5月，京东支付的支付验证方式支持了指纹验证，同时在京东商城体系内支持小额免密支付。

2016年8月，向合作商户App全面推广集成京东支付SDK，让用户在京东体系之外也能使用京东支付。同时，将手机京东App、京东金融App扫一扫业务接入京东支付SDK。用户可以用手机京东App和京东金融App完成线下扫码付款。

2017年618前夕，对京东App移动收银台全新改版，将京东支付在收银台内直接展开显示，更直观地将支付方式信息呈现给用户，降低了用户的操作成本，强化用户对京东支付品牌的认知。

2017年，京东金融作为服务于金融行业的科技公司，与中国银联进行深度合作，京东支付也支持了银联二维码，用户可以在银联商户扫描银联的二维码进行支付。

2. 背后的技术

对于京东支付来说，底层有用户卡中心系统、实名系统、支付密码系统、白条系统、小金库系统、钱包余额系统、钢镚系统等。当某个虚拟资产系统性能有问题时，京东支付要做到有开关实现下线功能，而当系统恢复时实现上线功能。面对618的海量洪峰及复杂多变的支付业务，如何高效应对？

收银台前置系统预热缓存方案。系统在用户每进入一次收银台前置系统时，就需要查询7次业务系统，如图2-21所示。

收银台前置系统可以一次开启多个线程并发的查询。虽然性能问题得到了解决，但是缺点也很明显：

◆ 响应时间长。

收银台的最终响应时间是以最长的调用为准。例如调用系统中的

的银行卡系统耗时60ms，其他的系统只耗时10ms，那么收银台前置的性能耗时就是60ms。

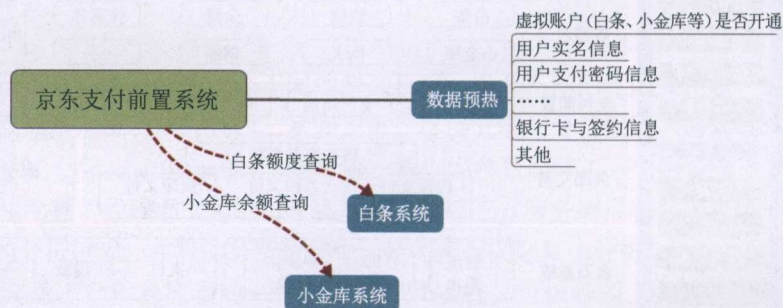


图2-21 收银台前置系统预热缓存方案

◆ **系统可用性降低。**如果下游每个系统的可用性是0.99，那么收银台前置系统的系统可用性就降低到了 $0.99 \times 0.99 \times 0.99 \times 0.99 \times 0.99 \times 0.99 \times 0.99 = 0.93$ 。

◆ **系统资源消耗较多。**既然一次请求就要开启7个子线程去查询下游系统的数据，那么面对前端收银台巨大的请求量，收银台前置系统需要有大量的机器来应对大的并发。

为了解决以上问题，团队通过分析现状，观察到可以将用户数据分为几类：易变数据，即变化频率高无法放到猎户数据中心的数据，比如白条额度，用户每消费一笔数据都会变化；非易变数据，即变化频率低可以放到猎户中心的数据，比如用户卡信息，用户添加一次后一般很少再添加；固定数据，即几乎不可能发生变化的数据，比如用户实名数据，用户实名后几乎不可能变动。

结合实际情况，解决方案就很简单了：将非易变数据与固定数据存储到缓存中提供给收银台前置系统进行查询，而对易变数据则由收银台前置系统继续调用下游业务系统进行查询。

改造后，收银台前置系统依赖的非易变数据与固定数据都做了缓存，系统绝大部分的请求会直接通过缓存解决，只有易变数据才会真正进行查询处理。收银台前置系统的可用性由此大大提高了。

移动支付网关方案。对于互联网公司来说，App发版节奏很快，慢到一个月一版，快到一周或半个月一版。这么多版本怎么管理，Android和iOS实现差异性怎么解决？为了解决这个问题，团队引入了移动网关，按照版本等维度使客户端的请求能发送到不同的服务器上。特别是当新功能上线时，不可能将所有的机器一股脑地全部部署完成。在保障不影响老版本的基础上，对新功能进行灰度验证，系统架构如图2-22所示。

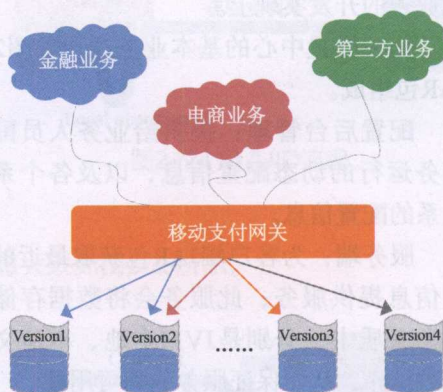


图2-22 移动支付网关系统架构

2.3.2 交易系统

身处崇尚创新、追求卓越的团队氛围下，京东支付开发团队本着对优良用户体验的无限渴望使得以多中心、高可用为核心架构思想的交易系统应运而生——称为金石系统。

金石系统的主导思想是尽量不强依赖于任何系统，包括数据库、缓存等存储介质；同时对于机房的可用性也做了全面的容灾策略，实现了交易系统的多中心架构。另外，主流程非常简洁，本着能异步就异步的基本思想，尽量降低系统运行风险，提升系统的自愈能力，作为未来5~10年的核心底层服务，为整个京东支付保驾护航。

金石系统支持网银支付机构的全部交易；囊括快捷、网关、白条、小金库、钱包余额、线下充值、预付费卡、支付营销等支付工具；支撑着收单、代收、会员、预授权、退款等交易产品；推动着账务、清结算、交易记录、企业站等业务系统的正常扭转，是支付



链路中非常重要的一环；承上启下，记载着支付过程中产生的支付数据，作为用户支付行为的重要凭证，支撑着每年的多次大促活动；对商城、金融、全球购、对外收单等不同商业主体提供高可用的底层支付交易服务。下面介绍系统的核心基础架构。

配置动态本地化。系统的运行态往往存在很多种，不是一成不变的，很多时候需要根据业务配置不同参数，从而走不同的业务逻辑。最初系统基本都采用Spring配置文件的方式来实现，将动态配置抽象到Properties文件中，基本能满足大多数系统的需求。但是在业务爆炸式发展的互联网业务系统中就不太适合了，会出现非常频繁的业务上线服务重启工作的出现，这样会对线上的高可用平稳运行带来很大的挑战，使得业务的上线流程比较零散，没有办法将其抽象为一个标准的流程，并且需要各个系统的开发人员来配合汹涌来潮的业务上线过程。

在这样的业务场景下，一个统一的、抽象的配置中心系统就显得非常重要，这样可以将业务的上线流程与各个系统解耦；开发人员不再需要天天配合业务上线，只需要业务运营人员在统一配置中心将相关信息配置好，各个系统就会自动获取到自己关心的配置信息，从而自动识别到新业务上线，完全解放开发人员，使其将更大的精力放到架构的升级及业务的开发实现上。

统一配置中心的基本业务逻辑如图2-23所示，主要由配置后台管理、服务端及客户端JAR包组成。

配置后台管理：提供给业务人员配置业务运行的动态配置信息，以及各个系统关系的配置信息。

服务端：为客户端JAR包获取最近的配置信息提供服务；此服务会将数据存储到三个介质中，分别是JVM本地、本地文件及数据库，从而保证服务的高可用性。

客户端JAR包：主动拉取配置信息，并将最新的配置信息推送给业务系统，业务系统从而获取到最新的配置信息，以支持最新上线的业务逻辑。

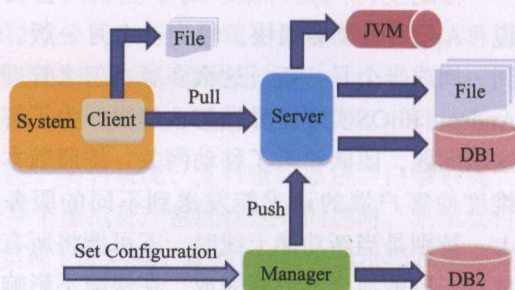


图2-23 统一配置中心

UUID本地化。业务系统基本都需要一个唯一的单号来标记业务内唯一的一条数据记录，这类似数据库中的主键，为的是保证业务内的全局唯一。通常采用数据库索引、数据库自增等数据库方式来实现，因为此方式强依赖数据库，所以数据库故障将带来无法预估的灾难。

核心交易系统依赖一个强依赖SQL Server数据库的UUID生成器服务。某次公司内部的一个Windows集群宕机，不幸的是UUID服务强依赖的数据库在此集群上，这就导致了服务挂起，从而导致核心交易挂起，最终导致整个线上交易挂起。更不幸的是，问题无法在短时间解决，整个挂起持续两个多小时，最终给公司带来了很大的损失。

有了这样的经历后，无依赖的、高可用的本地化UUID项目启动，该项目采用本地JAR包不依赖任何RPC形式生成UUID，不再强依赖数据库。主要是采用实例号本地化，并将实

例号打入到单号中，从而保证业务内单号以不重复策略来生成。主要实现流程如下：

◆ 系统初次启动时（本地UUID文件不存在），调用注册中心获取实例号，最多获取5次，如果5次都没有获取到，直接抛出异常，阻止实例启动，从而不对外提供服务。

◆ 在系统初次启动的时候，获取一次实例号，获取成功后，就跟UUID注册中心没有关系了，注册中心挂起，对应用没有任何影响。

◆ 首次启动获取到的实例号会保存到每个实例的固定文件夹下的UUID文件中，以后这个实例就用这个实例号，其他的实例不会再用这个实例号，从而保证各个实例间创建的UUID不重复。

◆ 剩下的就是保证一个实例下的单号不重复的问题了，其实现方式有很多，这里采用的是Linux时间戳^[1]方式实现：获取到秒，并根据业务的TPS每一秒钟设置一个实例。具体流程如图2-24所示。

数据持久化容灾。业务数据最终都需要落地，都需要持久化。业务数据的持久化最终肯定都需要落到数据库上，因此数据库就成为了整个系统最后的单点服务，而且非常重要，不能出一点问题，一旦出问题往往就是致命的。因此，取消数据库的强依赖是高可用系统的重中之重。

取消数据库强依赖的核心思想是：前端添加缓存降级，后端添加数据异步还原机制，以保证数据的最终一致性。金石系统秉承此核心思想，实现了一套不强依赖数据的数据存储策略，核心思想大致如图2-25所示。

◆ 接入JIMDB/R2M双备份缓存，同步写一个，异步写一个，主从可随时切换，从而实现尽量保证缓存数据的完整性；达到短时间数据库故障业务无感知的目标；将查询全量存入缓存，降低数据库压力。

◆ 采用JMQL做数据库容灾降级数据回写机制，并对JMQL做容灾降级，接入MQSender，尽量保证异步消息的高可用性，从而保证DB回写机制的高可用性。



图2-24 UUID流程

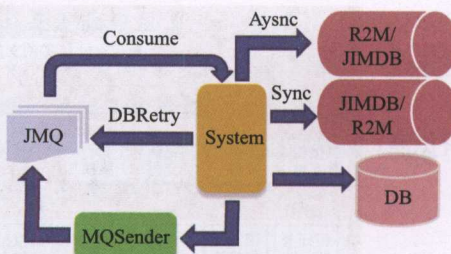


图2-25 数据持久化容灾

[1] 采用 Linux 的 NTPD 服务来保证多实例间的时间尽量同步。有时 NTPD 服务异常也会导致服务时间不一致，为了解决此问题，各个实例记录了当前实例已经用过的最新时间，从系统获取的时间如果小于用过的时间，那就用用过的最新时间加 1s 来替代，并发出报警；此时开发人员及 PE 将介入，并排查问题，最终将其同步为一致的最新时间。



2.3.3 自动网关

1. 支付网关架构和支付业务流程简介

◆ 支付网关的架构模块体系如图2-26所示。

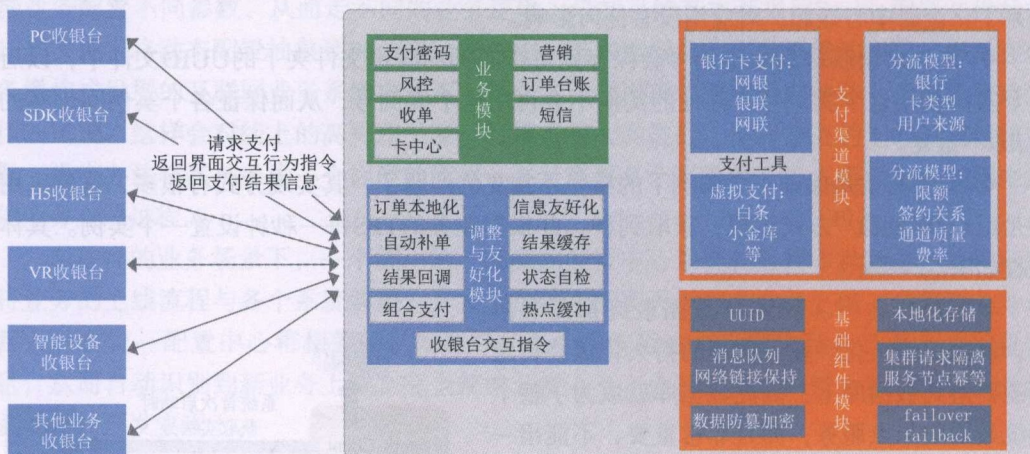


图2-26 支付网关的架构模块体系

按照功能主要分为业务模块、支付渠道模块、高速与友好化模块、基础组件模块。各个功能模块的组织谱系关系如图2-27所示。

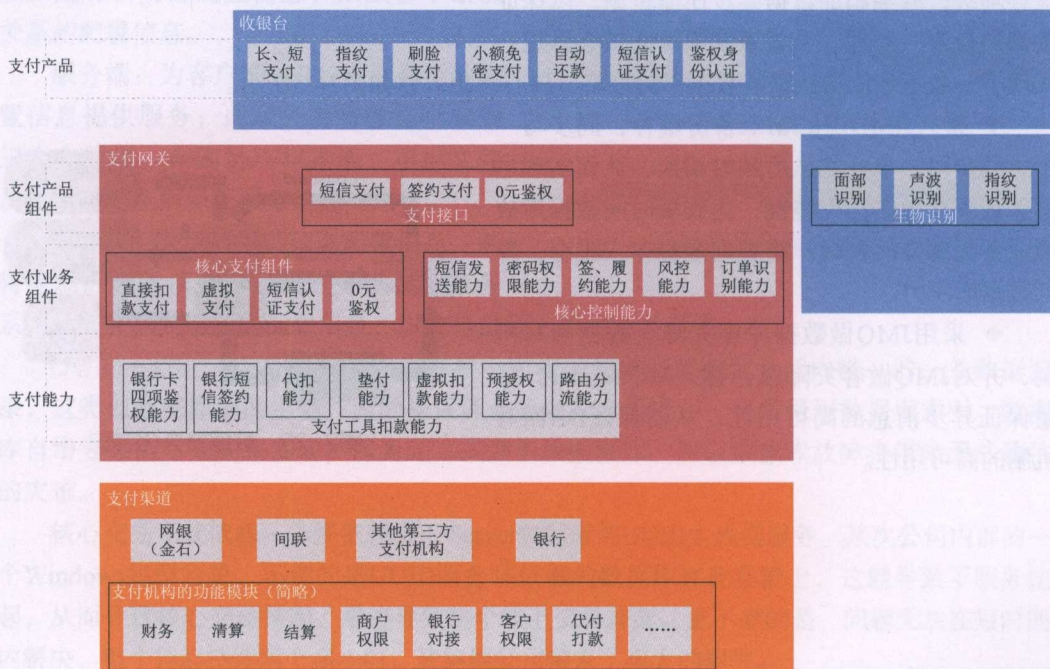


图2-27 各个功能模块的组织谱系关系

◆ 支付渠道。支付渠道（以下简称渠道）主要为支付网关（以下简称网关）提供了支

付工具的扣款能力，网关综合各渠道的扣款能力屏蔽机构间的差异，生成核心支付组件。再结合核心的业务控制能力（风控、路由、商品订单）进行流程化封装生成对支付接口。收银台增添一些先决认证条件，如生物识别、手机认证（短信、尾号）、SSO登录，设备认证通过后调用网关的支付接口进行支付。

2. 平行迁移

平行迁移功能是做故障迁移、流量定位转移的工具。例如，当实例1故障时，如图2-28所示。

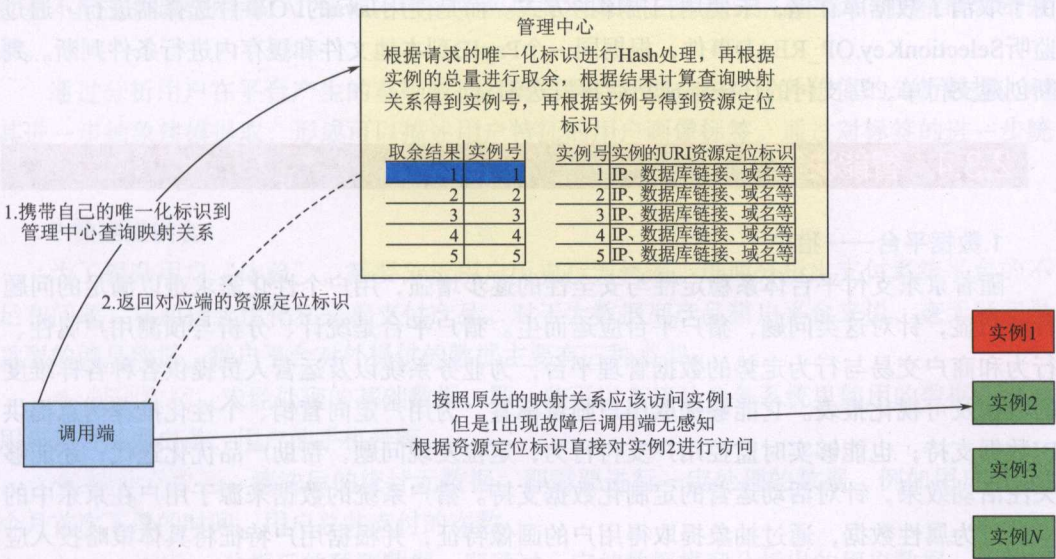


图2-28 实例1故障

◆ **映射关系。**调用端并不是每次都到管理中心获取映射关系，正常情况下调用端只在第一次系统启动时到管理中心获取对端的URI并记录到本地，只要对端正常就一直会访问。实际上调用端有开关功能，控制出现异常时查询还是每次都查询。

◆ **注册中心。**如果每次都查询管理中心，那么管理中心的性能如何保证呢？目前采用的是纯本地JVM的K-V类型Map + ReentrantReadWriteLock+数据库进行解决。

3. 本地化存储

◆ **一般消息性存储。**即出现异常时把要对外发送的消息先存储到本地，然后单独再起线程向原来的接受方进行传输。由于目前应用场景主要面向消息队列，已经逐渐被技术部门统一研发的MQSender所取代，是对消息队列在客户端的Failover机制的一种扩充。如果要自己实现，单就存储而言，采用MappedByteBuffer做内存和刷盘工具+ ReentrantReadWriteLock进行线程隔离就能满足需求，这里不再赘述。

◆ **有顺序保障的结构性存储。**如同一笔支付需要创建支付单（类似财务的应收概念）、写支付结果（类似财务的实收概念）两个动作。业务上要顺序发生并且必须要用数据库进行持久化存储。问题是创建支付单、写支付结果这两个动作在实际流程中因为中间涉及用户交互，延时掉单等问题往往存在支付结果先有，而创建支付单延时的情况；或者创建支付单很



久以后，支付结果才通过别的方式写入（通过银行发异步接口回调，对账单核对）。

首先，交易系统层的同事已经把支付结果和支付单放到不同的表内，做嵌入操作而不是更新数据。其次，如果入库操作出现异常，表数据也会首先存入缓存，等数据库情况变好后再调度入库。相当于一笔支付的支付单、支付结果都存在缓存或数据库时，再发起下级非实时业务。

而在支付网关的场景是：调用交易系统出现网络失败，在写入延时较长的情况下先断开与交易系统的交互，自行发送保存创建支付单和支付结果存入到缓存和本地文件。这是由于取消了数据库存储，未使用扫描库的方式。而是使用Java的I/O事件选择器进行。通过监听SelectionKey.OP_READ事件，根据同一个PayID到本地文件和缓存内进行条件判断。判断创建支付单、写支付的任务都成功后再发送消息。

2.3.4 智能决策

1. 数据平台——猎户

随着京东支付平台体系稳定性与安全性的逐步增强，用户个性化需求难以满足的问题日渐凸显，针对这类问题，猎户平台应运而生。猎户平台是统计、分析与预测用户属性、行为和商户交易与行为走势的数据管理平台，为业务系统以及运营人员提供各种各样维度的数据及可视化报表。它能够提取用户画像标签，为用户定向营销、个性化推荐信息提供BI数据支持；也能够实时监控用户支付行为，定位发现问题，帮助产品优化迭代；还能够关注活动效果，针对活动运营的定制化数据支持。猎户系统的数据来源于用户在京东中的平台行为属性数据，通过抽象提取得用户的画像特征，并根据用户特征将具体策略投入应用，其构建及应用如图2-29所示。

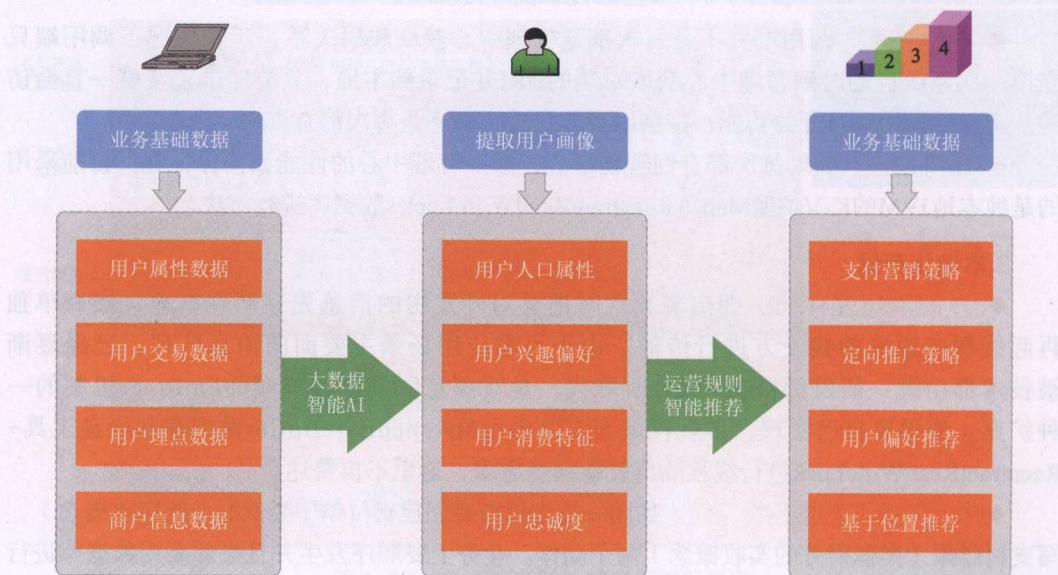


图2-29 数据构建以及使用

针对支付营销，猎户平台为营销系统提供用户支付属性标签，帮助营销系统定位目标用户，进而精准营销；收集汇总营销活动推行情况，跟踪用户使用营销优惠的后续支付行为，提炼更为精准的营销敏感度标签和营销价值标签，帮助营销系统创造更大价值。

针对LBS位置服务，猎户平台提供具象化的数据、定向搜索展示地图信息、承载LBS运营活动等功能。它根据用户位置、过往偏好、支付习惯等因素，为用户个性化地展示推荐商家及优惠，并能够输出到商户运营管理后台，方便运营实时监测重要数据指标。

针对资源位定向分发，猎户平台利用丰富用户数据资源，抽象出定向用户特征标签，针对不同支付属性的用户，采取不同的资源位展示策略，达到“信息找人”的精准营销效果。

通过分析用户在平台产生的基础数据，猎户平台发现一些有业务价值的分析数据，对其进行进一步抽象建模提取，形成可以描述用户特征的用户画像标签。通过对标签的进一步统计分析，得到具有群体性特征的统计报表，有助于提升业务效果。

2. 数据提供者

为了提升用户“体验”，需要分析用户历史行为轨迹，进而分析出支付系统平台的不足和问题，从而持续优化和完善支付产品。对于大数据型产品猎户平台来说，毫无疑问最重要的就是数据。猎户平台对外提供的数据主要有三种类型：

◆ **基础数据**。未经处理的基础数据，即一些可以直接从业务系统里使用的数据。例如用户是否开通白条、用户的手机号等。

◆ **统计元数据**。清洗后的统计元数据，即需要进行一定处理的数据。例如用户最近3个月首次下单的时间，用户总共支付的次数。

◆ **预测数据**。分析后的预测数据，即通过一定的数据模型分析出的用户数据。例如用户最近1个月有90%的可能性还会下一单。

需要说明的是，这三种类型的数据不是绝对的，而是相对的。有可能猎户平台接收到业务系统的数据已经是简单处理后的了，所以对于业务系统来说，这种数据可能就叫作统计元数据，对猎户平台来说就是基础数据。例如，某个用户绑定了三张银行卡，业务系统需要进行简单的统计卡数量规则。对于业务系统来说，这种数据叫作统计元数据；相对地，猎户平台接受到这个用户绑定了三张银行卡的数据后可以直接对外输出使用，不用再做任何处理，对于猎户平台来说这种数据就叫作基础数据。三种数据类型只是一个称呼问题，为了使系统能够更清晰地区分数据现状。

着重说明一下，有时有可能系统的数据建模与报表趋势发生了改变，导致需要将一些已经分析过的猎户平台中的数据进行再次分析，从而得出新的数据与数据模型。

与数据类型对应，猎户平台架构如图2-30所示。

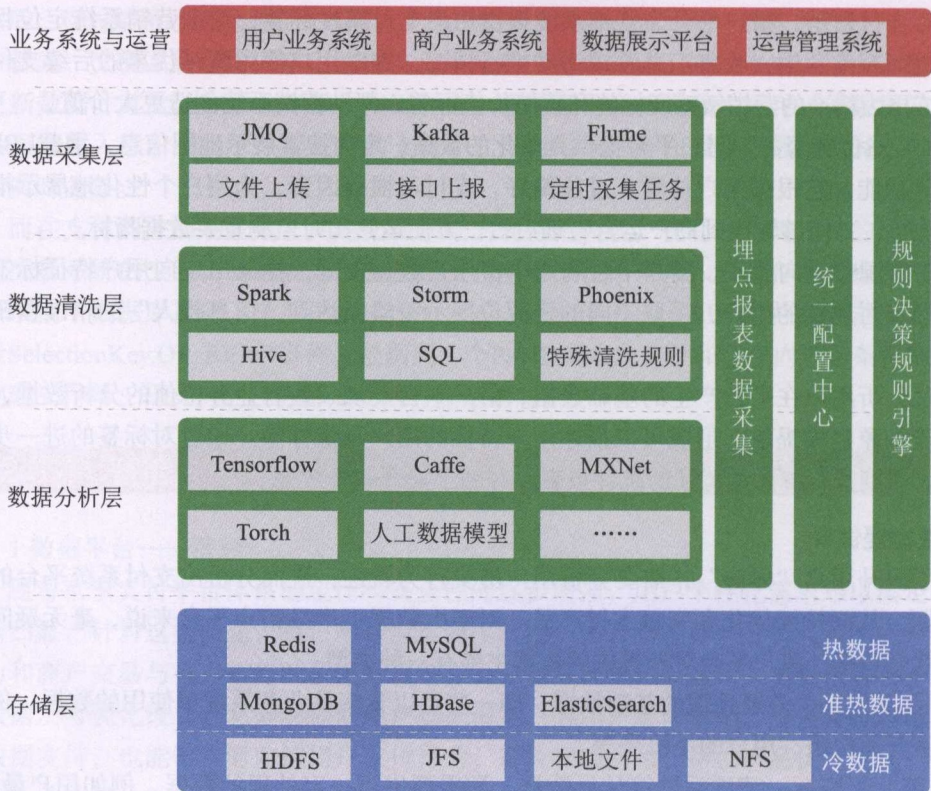


图2-30 猎户平台架构

3.数据清洗模块

数据上报分为业务上报模式和主动采集模式。业务上报模式是指业务系统主动将数据传输给数据采集平台，可以采用的数据上报方式有：接口调用、JMS、Kafka、文件上传等。主动采集模式是指数据采集平台主动去业务系统进行数据抓取，可以采用的方式有：关系型数据库Binlog复制、应用服务器增加日志Agent采集、文件抓取、爬虫等。这两种模式各有优缺点，要根据具体情况来设计数据采取的方式与方法。

数据清洗模块是将接收到的数据转换为实际所需要维度的数据。例如，猎户平台接收到了用户的卡数据，清洗模块结合存储的存量卡数据，统计出用户的绑卡数量。清洗模块里面的难点很大一部分在于防重性，执行过的数据不能被再次重复执行。在Spark Streaming和Storm中一般是指“Exactly Once”，读者可以参考其他资料学习研究，这里着重介绍猎户平台的处理方式。

猎户平台通过Redis中的Bitmap实现了BloomFilter算法来完成Exactly Once里的去重操作。因为数据量较大，所以需要构建多个Redis的Bitmap通过Hash取模的方式进行分散散列存储。Redis的Bitmap只支持232大小，对应到内存是512MB。映射到具体数据万分之一的误判率，512MB可以放下2亿条左右的数据。因为数据量是一直在增长的，所以Bitmap最终会被放满，所以在实际使用过程中，要考虑到一个具体情况就是Bitmap的有效期设置问

题。猎户平台最终采用的是如下方案：

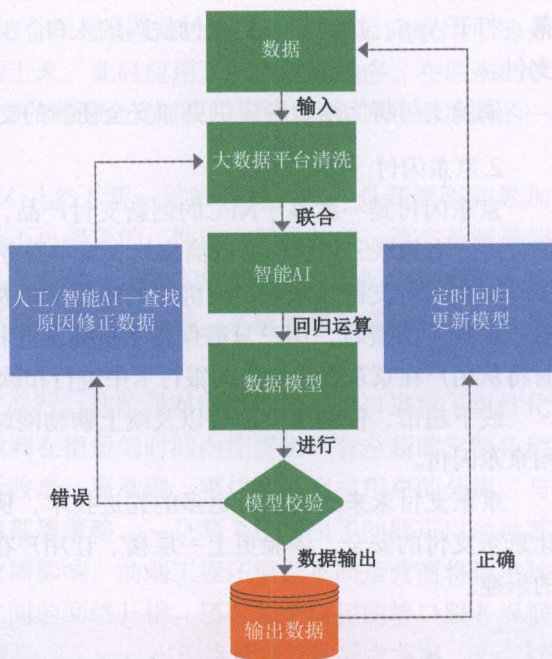
- ◆ 做双份Redis Bitmap缓存。
- ◆ 如果失效时间是1天，那么设置两份缓存的失效时间都是2天，但是两份缓存是按天间隔使用的。这样就能实现每天都有失效的缓存，每天也都有创建的缓存。但是缺点也很明显，本来预期需要缓存一天的数据，现在却每天需要创建两份2天缓存的Bitmap。

4.数据分析模块

数据分析模块的重点是建立各种各样的模型。猎户平台需要将统计整理好的数据通过各种各样的算法协同工作，最终生成需要的模型。得益于京东大数据部门，基于算法与模型的平台，数据模型的分析工作可以轻松地展开。用户画像分析是一个持续的过程，以各种数据为基础，通过建模系统进行分析得出模型，之后结合数据进行模型验证，如图2-31所示。

无论得出的模型正确与否都要继续分析，所以模型分析的工作量很大。不过好在模型是周期性的，只要对正确的模型按照其模型周期进行更新即可。

当数据离线分析与数据实时使用时，规则引擎起着关键作用。理论上数据离线分析准备好，线上实时系统直接使用是最好的模式。但是现实系统中毕竟存在着各种各样的需求，实际情况还是要使用规则引擎对离线分析后的数据进行转换后进行输出。



2.3.5 支付未来

1.刷脸支付

支付过程中对用户身份的验证是最重要的安全保障，传统的验证方式是通过账户、密码或者短信验证码；随着人脸识别技术的成熟和人们对支付便捷安全性需求的提升，能否把人脸作为一种识别和验证方式成为下一个支付创新的风口。

第一，人脸是唯一不需要用户有其他特殊配合就可以采集到的生物特征信息。通过人脸识别身份，符合人的视觉识别经验，容易被使用者接受。

第二，京东的人脸识别技术是由京东智能识别实验室自己研发的技术。京东人脸识别技术从启动到达到业界顶尖水平，通过内部场景的不断打磨，采用深度学习算法，对千万



级数据进行机器学习、模型训练，算法识别准确率和业界顶尖水平持平。从安全性来讲，目前已经超越市面上商业公司，可以有效防范视频、照片、面具等攻击手段。

基于以上两点，推出了基于人脸识别技术的新型支付方式——刷脸支付。刷脸支付将用户面部信息与支付系统相关联，通过拍照把获取的图像信息与数据库中事先采集的存储信息进行比对来完成用户身份的识别和认证，替代了之前通过账户、密码及短信验证码等方式对身份的识别和认证。

从用户体验上来说，和扫码付、付款码相比，刷脸支付时用户无需掏出手机、点亮屏幕、打开App、找到扫一扫和付款码的入口，只需要在屏幕前露露脸，确认付款信息即可支付。

刷脸支付将为消费者提供更加安全便捷的支付方式。

2. 京东闪付

京东闪付是一款基于NFC的创新支付产品，能够在无网条件下使用，具有快捷、安全等优点。有银联云闪付标志的商家均支持本服务。

京东闪付支持京东账户内绑定的银行卡消费，消费时用户可以享受银联和京东双重优惠。在线下消费时，用户只需保持手机亮屏并将手机贴近银联POS机即可完成支付，交易时将从用户在京东账户绑定的银行卡中进行扣款。

线下超市、便利店、酒店以及线上购物网站、应用，只要支持云闪付的地方都可以使用京东闪付。

京东支付未来还会接入更多的先进技术，例如虹膜、声纹、掌纹、蓝牙等相关技术，让京东支付的安全、体验更上一层楼、让用户在京东支付上可以享受到更多的优惠、更多的乐趣。



JDS：支付产品部负责人，负责京东支付的产品研发工作。

2.4 白条业务——先消费，后付款

2.4.1 智能授信

1. 白条授信在618中的演变历史

白条智能授信平台演变大致有三个阶段：第一阶段，2014年6月搭建白条激活系统的第一版，当时系统设计很简单，一个前端应用和一个后端应用，数据库是单库单表。业务流程也比较简单，仅通过银行卡鉴权完成实名认证，后续会走授信策略完成最终的用户激活开户流程。

之后随着业务的快速发现，开展了校园白条、园区白条、乡村白条等，以及后续的新、社交等业务流程，增加了许多差异化的激活渠道和应用场景。随着渠道越来越多，流

程越来越复杂，这时带来以下问题：第一是入口多、太分散，难以统一管理；第二是每个渠道之间界限难以界定，用户在后期管理中面临各种权限划分不清的问题；第三是用户体验太差，感觉整个开户体验很复杂、耗时长。面对以上问题，各渠道之间的协调和交互变得越来越重要，这时候激活路由系统孕育而生。这些差异化的激活渠道针对不同人群，满足业务、风控，以及产品体验上差异化的需求，同时还最大化用户体验，最好的情况是用户只需点击一次按钮即可完成整个激活流程。

激活路由系统承担着对用户的分群和路由工作，在用户做白条开户业务的前期就已经开始了工作，读取多个渠道的路由规则，结合渠道优先级完成相关计算后完成用户的分流，最后引导当前用户到最合适的流程页面上来。此时应用系统也越来越多，在原来的单一系统上进行了垂直拆分，把一个应用按业务维度拆分成多了应用，数据库也由原来的一个SQL Server迁移到多个MySQL数据库。

业务的快速发展是难以预料的，随之又引来了新一轮的问题：第一是新流程频繁加入，并引入多种校验逻辑，相互间耦合还是不可避免的，性能也越来越差；第二是海量的需求如洪水般涌来，团队成员即便已经超负荷工作，工作进度依然无法令业务方满意；第三是依赖的外部接口越来越多且不稳定，系统内部互相依赖，系统频繁出现问题，而且定位时间长。

针对以上种种问题，新的升级改造势在必行。这阶段对所有的原子接口进行了组件化拆分和组装，并增加了可配置平台，以便做到在很短的时间内配置出一套全新的定制化激活流程。同时对激活路由进行了多线程并行改造，更准确、更快速地完成用户的分流。与此同时，所有的应用子系统全部采用多机房部署策略，一旦某个机房出了问题可以通过其他机房来保持业务正常运行，免受单机房故障影响。前端工程还根据不同运营商将请求转发到相对应的机房进行处理，减少跨运营之间的网络开销。还有每个应用的接口服务根据业务的重要等级也通过JSF采用不同的分组来隔离，一旦出现流量高峰和异常情况，也能优先保证核心服务不受影响。最后数据库也做了水平方面的拆分，实现了多库多表、主从互备，从而满足海量数据的应用需求和性能需求。

2. 技术架构

智能授信平台不是一个单一的应用系统，是由24个核心应用子系统组成的（这里还不算依赖的第三方应用），共提供了200多个接口服务，10多个独立业务数据库（业务维度），涉及MySQL、MongoDB、HBase，总计数百张业务表，依赖的缓存工具有JIMDB和R2M。数据监控由1000多个核心技术监控点和100多个核心业务监控项，以及400多张实时数据图表组成。总体概括可包含以下三部分：激活路由、组件化和配置平台、监控分析系统。整体技术架构如图2-32所示。

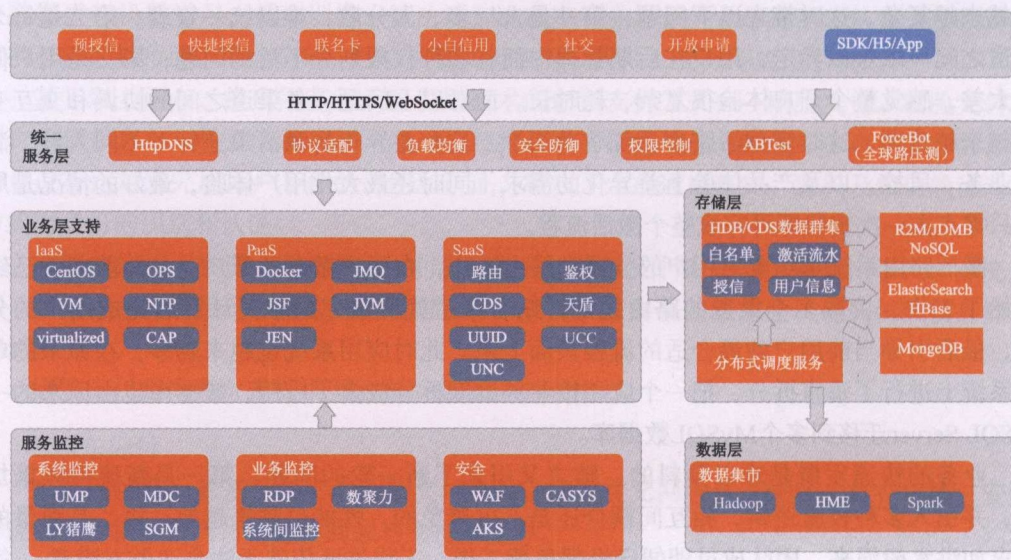


图2-32 白条业务技术架构图

3. 激活路由系统

激活路由系统就像整个生态系统的门卫和引路人，非常重要，所有的请求都会先到达这里，这里会有两个环节：一类是名单类的校验逻辑，一类是用户信息相关的业务规则校验逻辑。后一类检验的早期版本是放到路由后面进行的，因为路由系统对目前十多个激活渠道的规则都要进行筛选，所以对系统性能是一大考验，也影响用户体验。白条系统改进了原有的调用次序，改成了多线程并行执行，实时筛选黑白名单和准入规则，并将结果存入数据库，最后通过优先级规则和数据库的结果计算后得到一个最终的结果。

整个过程依赖很多业务规则和第三方RPC接口的调用，但整体性能控制在TP99为100ms左右。在代码结构上也进行了大的重构优化，采用的ExecutorService和模板方法设计模式等避免了原来大量if-else的方式，整体结构清晰、可读性强。整个流程经过的每一个环节的快照都通过MQ异步存储到大数据中，用于后续的查询、离线加工和分析。目前线上承载的激活渠道已达到15种之多。

4. 组件化和配置平台

随着业务的不断发展，越来越多的功能化模块被应用于内外部多个场景，不仅满足白条授信激活，还同时为金条、闪付、小白信用等多个业务提供基础服务。组件化的服务内部也细分了2个层次，最底层是原子粒度的服务，中间层是对原子服务的组合和封装，提供套装服务如图2-33所示。

用户信息及属性相关都持久化到数据库中，数据表结构采用窄表的设计思路，具有很大的扩展，同时采用缓存来提高性能，并且采用JIMDB和R2M双缓存的互备双写方案，其中一个出问题，可以通过ZooKeeper开关来控制，做到秒级的切换。

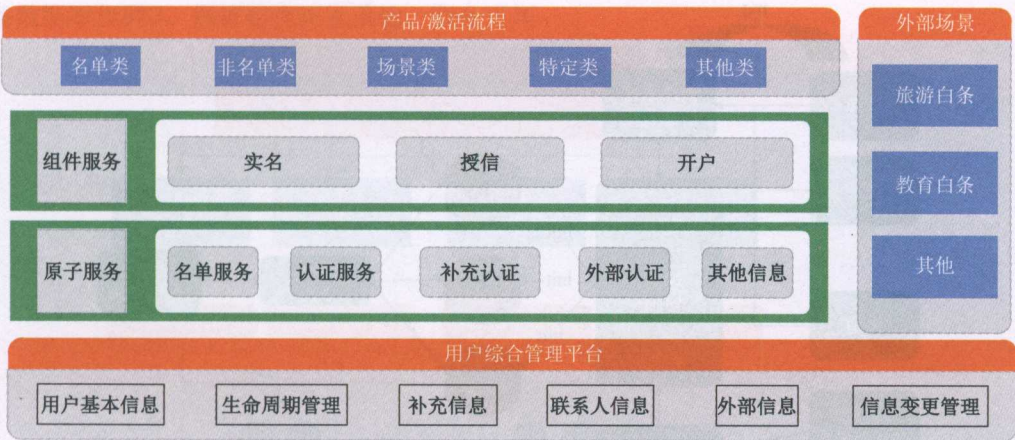


图2-33 组件化和配置平台

此外还有一套配置管理系统，用于配置差异化流程所需要的关键要素和功能节点。举个例子，之前要开发一套激活流程，由于定制化程度很高，依赖的外部数据等都需要按照新要求对接、测试上线等，整个过程需要多个开发人员全程参与，工期漫长，风险也很高，当业务需求增加或频繁变更时，人手又不够就只能排队了。现在真的可以说是分分钟完成一个业务需求的定制化开发，整个开发测试最快情况可在一天内完成，而且配置灵活应对不断更改的业务需求。配置管理系统细化到每个接口字段上，通过Map映射，灵活地对接上下游系统，甚至不用改一行代码。

2.4.2 便捷交易

2014年6月以来，白条在京东商城实现了对手机、电脑、家电等实物的线上交易，也逐步延展到虚拟类交易、外单交易、银行类合作支付、银联线下支付等支付业务类型。“白条走出去”使得白条交易流程对安全性、稳定性、高性能的要求越来越高，带来了更多技术和业务上的挑战。在2017年618期间白条交易平台计息系统调用量每分钟数百万，白条人口级展示调用量每分钟上千万，交易核心数据预热系统每分钟调用量上千万，交易营销规则匹配每分钟数亿次，交易支付成功率99.99%。

1. 业务服务治理

随着业务的不断发展，交易过程中依赖的服务、数据指标越来越繁杂，订单、账户、营销、用户数据等都不可或缺，那么如何加快系统响应速度，及时处理数据请求？

我们提取交易过程中使用的数据指标，并划分为两大类，即相对静态指标数据（如用户、账户）和动态指标数据（如订单、营销优惠）。根据业务特性的不同，我们对数据进行预热，数据计算处理进行前置，达到随用随取的目的，尽量减少交易过程中系统性能消耗。

白条预热系统结构如图2-34所示。

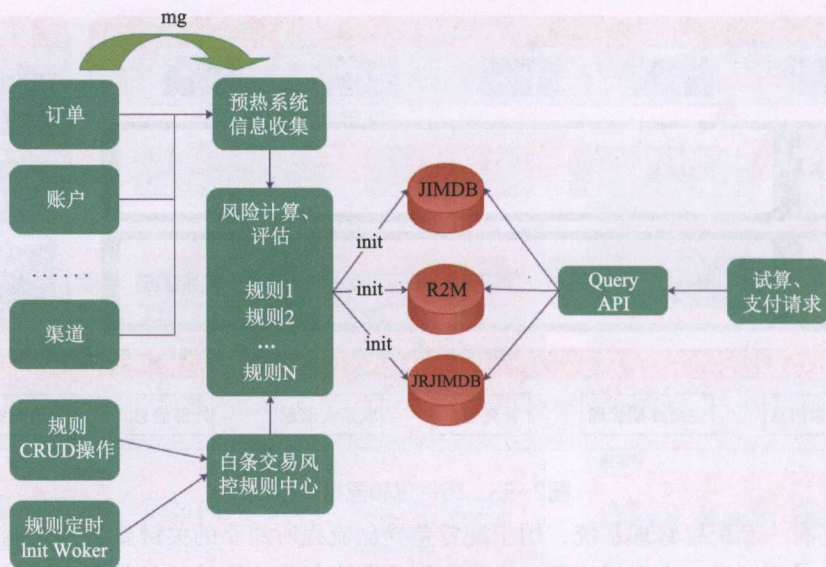


图2-34 白条预热系统结构

该系统结构满足如下要求：

- ◆ **数据时效性。**通过JMQ方式触达业务规则。相对静态数据，监听数据的MQ变化，实时预热到集群中。使数据指标动态化，如监听下单消息，进入交易环节，前置计算多项交易业务数据指标。
- ◆ **数据一致性。**对于相对静态数据，通过监听数据变化JMQ，同步数据预热，每日通过Worker进行数据同步，以保证最终一致性。
- ◆ **数据稳定性。**离线、实时计算结果存储于商城JIMDB、金融JIMDB、R2M中，多套数据存储灾备。
- ◆ **服务稳定性。**通过热备和流量负载，实时路由切换服务。

2.服务架构的高可用

对于交易系统而言，支付服务的持续可用是系统设计需要考虑的重中之重，哪怕一分钟的服务故障，损失的不仅仅是巨额的交易收入，更重要的是用户体验。

白条交易系统历经多次618，每年以200%+的速度增长，在系统并发量、稳定性、扩展性和灾备能力方面都要承担着巨大考验。

白条交易系统部署如图2-35所示。

业务层、数据访问层通过数据库的异地多活、多机房部署，然后基于JSF发布服务，再通过全局负载均衡，动态分配流量，当某台机器或某个机房发生故障时，可通过管理平台自助下线白条业务服务，保证业务请求正常梳理。

数据库作为整个交易链路中最重要的一份，数据落地不容有失，通过DB+（MySQL+NoSQL+分布式）同城双活的方式实现灾备。

白条核心数据库依赖金融自研MySQL集群，数据库采用分库本表，数据散列分布其中，而集群A、B同属生产级机房，数据实时同步，当集群A发生故障时，可自动切换到灾

备数据库集群B，保证数据正常落地、服务可用。

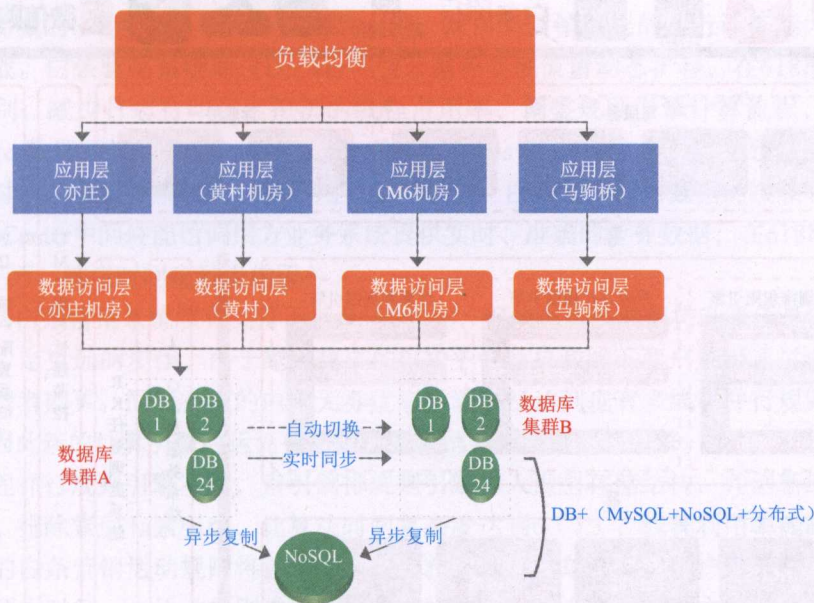


图2-35 白条交易系统部署

另外，热点数据的CRUD操作，都会通过消息的方式同步到NoSQL数据库，618期间查询量骤增，此时对于关系型数据库而言性能消耗主要源于查询，为了减少MySQL集群压力，视情况将高并发的热点数据查询切到NoSQL。

2.4.3 白条精细化营销

1.灵活多元化营销平台

与传统的金额优惠券不同的是，白条优惠券支持满减、直减、随机立减、N元购这些满足不同优惠力度的本金类优惠以及息费类优惠券。消费者在京东商城购买自营类商品时，使用京东券等京东商城优惠后，可以在收银台额外使用白条优惠券，最高支持24期分期手续费全免的优惠。这种优惠叠加，一经推出就非常受欢迎，之后随着京东金融消费者场景的扩展，全球购、京东到家、POP平台交易、金条、闪付等场景相继上线白条优惠券，给白条带来了成倍的交易量，同时也给白条营销系统带来了成倍的流量并发压力。

为了满足多场景的差异化营销需求，采用垂直拆分业务服务的架构解决不同场景的流量差异。通过JSF服务按场景划分业务集群，不同集群给不同业务收银台提供服务，实现了业务隔离和很好的扩展性。具体如图2-36所示。

在系统设计搭建时，将白条优惠券的领取、发放和使用等不同环节水平拆分，应对白条优惠券整点抢券、618秒杀等不同需求的流量冲击，通过JMQ异步服务实现抢券、秒杀等高并发请求的最终一致性保障。

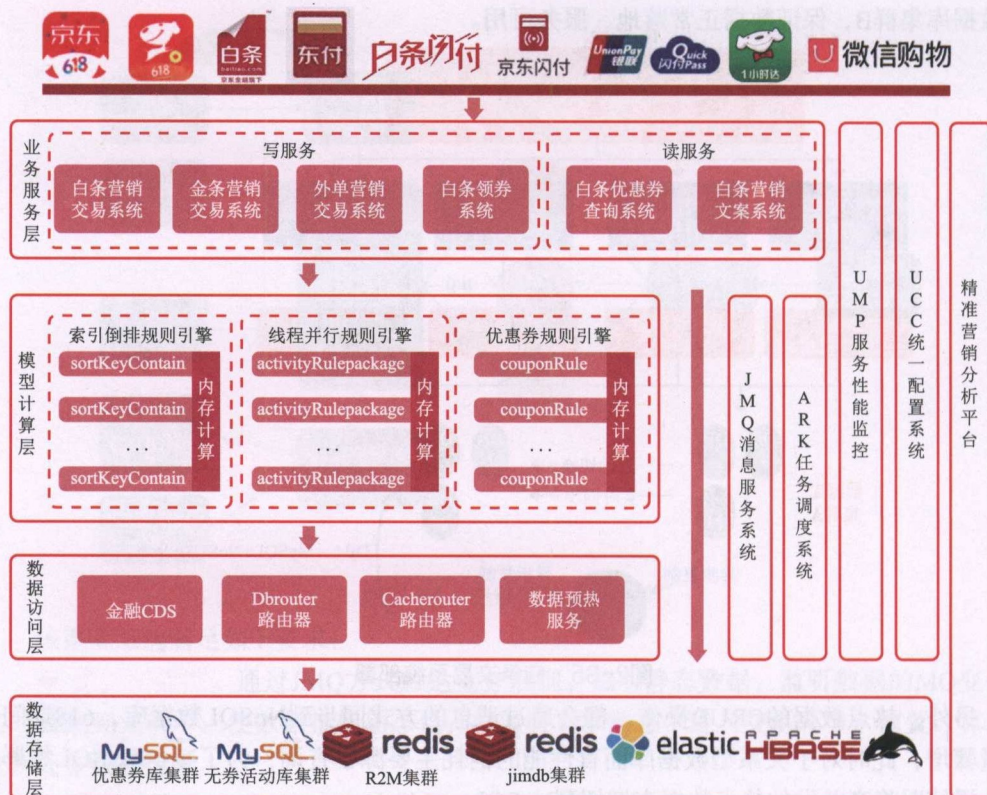


图2-36 白条营销系统架构

随着口碑的建立，白条营销诞生了另一个杀手级功能：白条无券优惠。消费者可以不领取优惠券，通过精准营销推荐平台预先计算该消费者是否可使用优惠，在支付环节自动生成订单的优惠。如此神奇的用户体验，背后的技术支撑可不简单：

- ◆ 基于ZooKeeper的UCC统一配置中心，让规则引擎计算平台的服务器及时准确地获取所有白条无券优惠活动的变更信息；
- ◆ 基于JVM虚拟机内存多线程并行计算架构的规则引擎计算服务，给百万级支付请求提供最优的白条无券优惠匹配；
- ◆ 在白条闪付等场景，用户整单退款后会将该订单使用的优惠退还给用户，并通过任务调度系统，维护白条营销优惠的库存增减。

可以看出，白条是非常注重用户体验的。白条营销也承载着白条用户体验很重要的流量获取工作。在京东商城商品详情页、搜索和购物车列表页、商品结算页、用户领券中心、微信手Q上，随处可以看到白条的优惠信息，这些白条优惠信息很方便地给用户购买提供参考，引导用户完成白条消费，也给白条营销系统带来了巨大的流量请求。

2.大促压力下的规则引擎优化

618的亿级流量请求促使系统架构一次次升级，加入基于Redis的R2M、JIMDB双缓存集群，针对不同网络运营部署多机房，加入ES索引服务以及CDS和DBRouter服务进行数

数据库垂直+水平拆分，实现读写分离。

京东商城的618流量是平日峰值的数十倍，对白条营销系统的并行计算能力提出了非常大的挑战。白条营销系统通过Docker弹性云服务实现流量动态扩容，在618期间降低日志输出级别，减少日志打印在虚拟机的线程占用率，调整规则引擎计算流程，并且引用CacheCenter数据预热服务。CacheCenter是基于Redis+Docker集群的数据预加载服务中间件，通过对京东商城订单数据、白条用户账户数据、白条用户行为数据的接收、加工、存储，CacheCenter中间件能给调用方业务系统提供实时、准确的业务数据，在618期间能提供QPS达到千万、响应时间毫秒级的服务。

现在的白条营销系统顺利支持了近几年的618，但快速迭代的架构升级还是满足不了白条走出去、走更远的步伐。由于越来越多的POP平台商户和线下商户选择通过白条优惠方式吸引消费者购买，同时在线的白条无券优惠活动已经达到现有多线程并行规则引擎的计算上限，因此新的计算引擎应运而生——白条营销索引倒排规则引擎。

与线程并行规则引擎不同，索引倒排规则引擎加入索引存储结构，并创新地加入区间排序索引、排除索引和索引组，让算法时间复杂度达到 $O(1)$ ，线程利用率提高1000倍。目前线上的白条营销活动规则将近50个，它们精细地将订单商品、用户行为和用户属性分类，通过预先对每一个活动规则进行索引数据初始化。一个活动规则之间可存在与、或、非关系组合，同样基于JVM虚拟机内存计算，及Redis数据预热双读双写服务，达到满足百万级商家百万级流量的性能要求。

3.比“你”更懂你——精准营销推荐

为了将场景下匹配的营销优惠主动推送给用户，实现千人千面的差异化营销，精准营销推荐平台应运而生。

精准营销推荐平台的核心部分是基于机器学习的数据挖掘决策树、逻辑回归、聚类等算法，目前白条精准营销数据分析因子元素属性上百个，针对新手礼包、快捷激活等场景的白条优惠券发放提供数据决策服务。618期间，用户白条优惠券使用率提高150%。



执笔人 JDS：支付产品部负责人，负责京东支付的产品研发工作。

第3章

移动互联

便捷个性，体验至上

忽如一夜春风来。人们的注意力正从PC快速转向手机，手机不仅是社交中心，也是电商中心。建立完善便捷的移动端购物体验，是京东走向下一个十年打造的重点。移动互联时代，京东已经在线。





3.1 超级流量入口——App

3.1.1 基础服务

1.JDReact三端融合开发平台

无线端开发痛点

京东的无线客户端在开发中经历过不少痛点，主要有如下几个方面：

◆ **效率低下**。因为无论是Android还是iOS，使用传统的原生开发都有一定的开发门槛。而且代码不能复用，这意味着任何一个业务都要在Android和iOS上各开发一次，测试和业务开发工作都不能复用。

◆ **性能比较差**。受限于WebView容器的一些瓶颈，传统的H5开发方式无论在页面加载还是在用户体验上，和原生应用相比均有较大的差距。

◆ **灵活性 不够**。传统原生开发意味着任何改动都需要发版，Android和iOS发版的成本都非常高，若遇到线上紧急问题，处理起来会非常麻烦。

◆ **业务接入困难**。因为Android和iOS平台有差异，很多业务代码和业务流程并不能复用，所以任何一种垂直业务接入App都较为困难，业务团队的开发、接入成本过高。

解决之道

团队也在反思到底需要一个什么样的平台来解决这些问题，然后在2015年注意到Facebook发布了ReactNative框架。简单来说，这是一种跨平台的移动应用开发框架。在当时它非常具有颠覆性，因为它最大的特点就是完全用JavaScript进行应用的开发，最终会渲染成原生的组件。对开发者来说，这意味着你提高了Web开发的效率，同时兼顾了原生的性能。因此，业界的吸引力非常大，该框架一经推出，就受到国外很多公司热捧，而Facebook自己也在使用。在国内，手机百度、微信手Q、京东App也很早就进行了开发。

基于RN框架的深度定制和二次开发

考虑到RN也有一些缺点，于是团队基于RN框架进行了深度定制和二次开发，逐步打造了符合京东业务的JDReact三端融合平台，如图3-1所示。

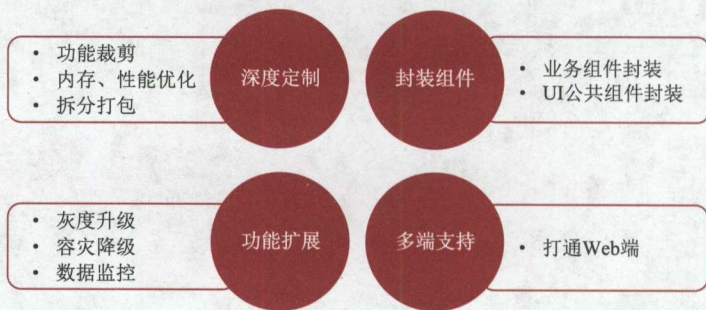


图3-1 JDReact三端融合平台

◆ **对RN的核心Base库进行裁剪和二次开发**。删减不需要的功能，修复性能、兼容性、稳定性的问题，同时也支持拆分打包。

◆ **在后端搭建一个功能支撑平台。**增加了灰度更新升级、数据监控以及容灾降级功能，这些对业务来说是非常重要的。

◆ **封装组件。**基于整个RN框架，结合京东的一些业务特点，团队封装了一套自己的业务组件，包括UI公共组件库。目的是让垂直业务开发者可以很快地使用框架进行业务开发，完全不用关心设计的样式和交互，做到快速接入业务。

◆ **打通Web端。**团队开发了一套RN框架向ReactJS转换的工具，可以实现一次代码编写，直接部署到Android、iOS和Web三端的功能。

2. 登录组件

随着移动互联网时代的到来，京东在无线领域的新业务迅猛增加，大量的优秀服务App化，用户对移动终端的账号登录提出了更高的要求。2015年之前，京东无线领域业务登录还处于非常发散的状态，导致了诸多问题。各业务分头使用验证密码等原子接口，自己设计登录序列，存在重复开发投入大、安全风险高、全公司范围内进行安全策略调整响应慢、密钥更新困难等问题。为了解决以上问题，2015年上海无线基础平台研发部推出“登录组件”，收拢无线领域登录入口，提高移动终端用户登录体验，加强用户账号安全及业务安全，重点面对智能终端的客户端业务，提供公司级统一的登录解决方案。

从图3-2可以看出，登录组件是一套登录鉴权体系，在登录鉴权体系中，票据（Ticket）是串联整个登录流程、验证用户身份以及确保通信信道安全的最重要部分。顾名思义，票据与团队日常生活所使用的公交车票、电影票等类似，使用者必须通过某种方式，如购买或申请来取得票据，之后即可利用此票据来享受某种服务。

登录组件化的优点

◆ **多样化的登录方式。**登录组件支持App登录注册、人脸登录、微信登录、QQ登录、授权登录、PC扫码登录、PC跨屏登录、M登录注册、M一键登录等。

◆ **用户体验好。**京东旗下的App可以互相拉起做授权登录，尽可能减少用户输入密码的次数，提高App的易用性。

◆ **账号及数据安全。**登录组件在整个登录流程中都是加密的，并且避免密码在网络上传输，保证用户账号安全，用的传输协议是私有的二进制协议，难破解。

◆ **集中化Session验证。**当用户更改密码或有安全风险时，需要剔除用户登录态票据。在接入团队登录组件之前，需要App的各个模块自己来处理，各个模块订阅MQ，然后剔除登录态；在接入团队登录组件后，只需登录组件服务订阅MQ，同时做剔除登录态操作。

◆ **易于更换密钥算法。**如果密钥泄露，可以定期更换密钥算法。京东商城App接入登录组件之前，如果要更换密钥，则各个模块都需要更改，各个模块服务都需要重启，涉及

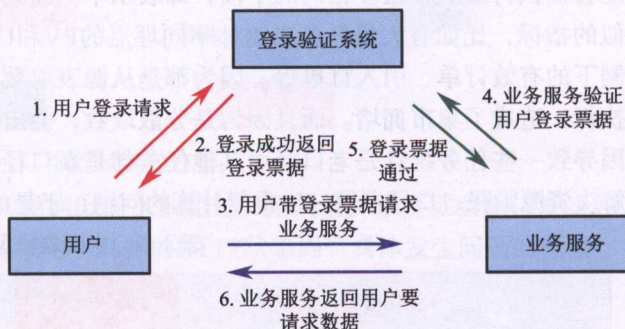


图3-2 登录组件典型工作示意图



的服务太多，根本无法做到同时重启。而接入登录组件之后，极大地改善了此问题。

◆ **有异地部署和容灾机制。**接入层、逻辑层、存储层都做了多机房部署和容灾。此外，在极端情况下，服务器具有过载保护能力，能够有效防止服务器雪崩，保证系统稳定运行。

◆ **完善的监控体系。**团队有完善的全链路监控体系，以稳定的服务保证系统及业务的正常稳定运行。

安全防刷多重保护

“羊毛党”的行为距离欺诈只有一步之遥，他们的存在严重破坏了促销活动的进行，侵占了活动的资源，使得正常的用户享受不到促销活动的实惠。而登录组件安全防刷就是对“羊毛党”的第一步拦截，它至关重要。登录组件安全防刷主要通过4层风控来限制“羊毛党”刷接口、恶意撞库、囤号。用户登录时，只有全部通过登录组件安全旁路、上海安全组黑名单、上游账号系统安全策略、京东集团风控四道门槛，才能成功登录。

3.基础数据

随着越来越多的用户通过手机访问京东购买商品，目前已有80%以上的订单来自手机端，用户在无线渠道的数据也成倍增长。伴随而来的是计算资源紧张，每天的用户、商品、订单的数据分析固化任务数有2000多个，集市的并行任务数常常在50多个，很多任务从排队到进入队列需要等待很久。与此同时，业务和口径也错综复杂。相同口径的计算可能会被执行多次，很多相同的字段，却表示不同新老口径的含义。大量的任务都在计算类似的指标，比如有大量任务在统计不同埋点的PV和UV数据，有大量的任务在统计不同限制下的有效订单、引入订单等。因为都是从源表取数，任务依赖满足后大量类似任务同时执行，造成了集市拥堵。而且因为是分散取数，会出现因为相关人员通知时间不一致等原因导致一些任务统计是老口径而其他任务却是新口径的状况，所以有必要提供一个方案来解决资源紧张、口径不统一、重复计算的问题，于是中间表出现了，中间表模型见表3-1。

表3-1 中间表模型

优化前	中间表	优化要求
并行任务多		提升查询效率
口径不统一		历史脚本通用
重复计算		可封装，不需要理解细节

中间表在解决上述问题的同时，还需要满足提升查询效率的要求，即必须方便落地执行，能够让历史脚本尽量可以通用，并且可以封装成黑盒，让使用者不需要理解其中的细节就能够使用。

3.1.2 技术支持

1.HTTP状态码：快速发现Web问题

无线业务目前已有450多个公网域名，基于业务模式和技术架构，大多数业务前端都采

用WebServer转发。虽然在用户接入层已有心跳探活机制来防止流量转发到问题机器上，但这只能探测WebService本身的状态，对于其后端真实处理业务的服务还无法准确探知。

京东作为一家大型互联网公司，对业务的监控是多维度、多方面的。无论是在最前端的用户接入，还是在后端的真实处理，技术人员都需要有很多监控平台来确保能及时发现问题。因为WebService作为一个承上启下的转发服务，所以团队不能只监控服务本身是否存活或者简单的连接数、并发数的数值，因此无线运维团队考虑从HTTP状态码入手进行监控。

作为运维团队，部署监控最根本的目的是迅速发现问题，同时也要考虑部署、开发是否简单，成本是否可控，所以团队首先把通过日志进行大数据分析的方法否决了。后来根据网页服务器本身的一些特性，决定利用网页服务器自带的功能来实现对错误HTTP状态码的转发，并利用开源软件NOSQL+Redis+MySQL来处理汇总数据，并发送告警，如图3-3所示。

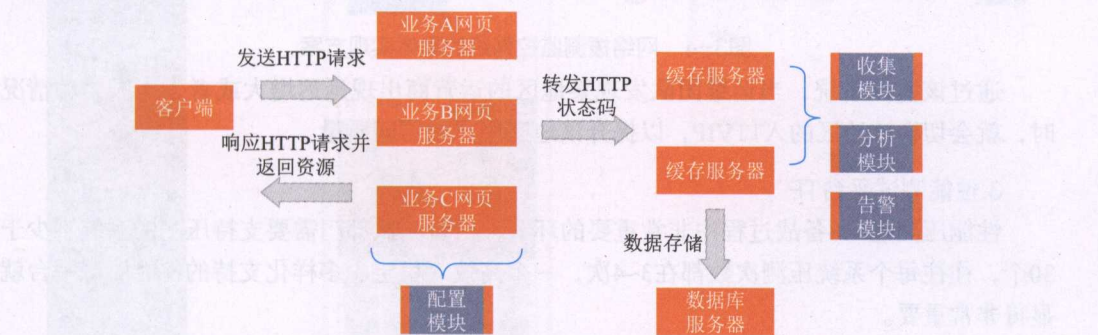


图3-3 状态码监控方案

当某个域名在指定时间单位出现50x和40x的次数超过一定阈值，系统就会默认发送告警。这样运维团队就能及时得知当时的域名、具体的HTTP状态码、具体发生问题的机器、当时单位时间内发生的次数以及请求的URI。

2.网络拨测：发现用户真实的网络问题

很多时候技术人员提交的用户反馈会使用一些笼统的描述词语，比如页面加载很慢、图片打不开等。虽然运维团队会回访部分用户了解情况并帮助他们定位具体问题，但这只是杯水车薪。在大部分用户的反馈中，除一些功能建议和功能Bug外，剩下的多数是网络连通质量的问题。

本着客户为先的精神，团队思考有没有一种更好的方法来及时发现用户真实的网络情况，比如当某地区的运营商出现网络质量问题的时候，系统可以及时切换，来提升用户体验。于是运维团队和LBS团队合作，开发了一个基于用户IP的网络拨测监控系统，具体实现方案如图3-4所示。

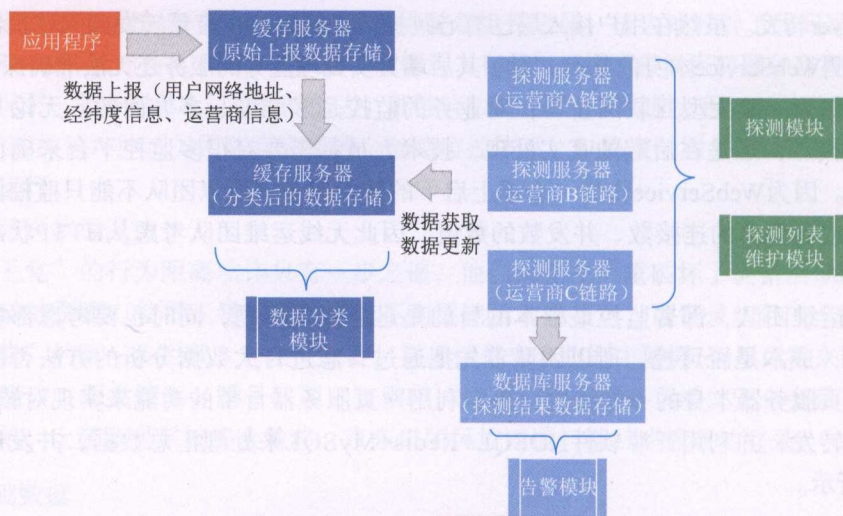


图3-4 网络拨测监控系统的具体实现方案

通过该监控系统，当运维团队发现某地区的运营商出现延迟增大或者丢弃频繁的情况时，就会切换该地区的入口VIP，以提升该地区用户的访问质量。

3.性能测试平台TP

性能压测是618备战过程中非常重要的环节，后台测试部门需要支持压测的系统不少于30个，往往每个系统压测次数都在3~4次，一个高效、稳定、多样化支持的性能测试平台就显得非常重要。

TP是京东后台测试与运维团队倾力打造的性能压测平台，该平台集成了Gatling、JMeter等主流性能测试工具，并加入京东独有的个性化元素，实现了以测试任务为主要维度的性能压测流程。目前平台已经能够完美支持HTTPS、JSF等接口的日常压测以及618压测。

性能压测平台的实现原理如图3-5所示。

为了让压测过程更人性化，平台功能也在不断优化和完善中，2017年再次简化了创建任务、执行测试任务的方式，同时加入和优化了对被压系统的监控视图，测试人员可以更为全面地观察测试发送的流量以及系统实际承受的流量，便于随时调整压测方案。

经过多次618备战的实际演练，性能压测平台已经逐渐确立了其自身在压测过程中的地位。目前包括研发人员单机调优、日常小规模压测、618大规模备战等都已经离不开TP。随着系统的逐步优化和不断新增功能，相信平台必将发挥其更大的价值。



图3-5 性能压测平台的实现原理

3.1.3 通天塔

1.千汇万状——敏捷快速搭建丰富的活动和频道

◆ **高效运营。**随着京东电商业务的发展，运营规模扩大，业务需求攀升，资源紧缺、重复开发、沟通效率低下等问题，暴露得愈发明显。基于现有大量业务系统的结构固定、业务需求相似度高等特点，从复用和优化资源配置的角度出发，研发一个搭建页面的可视化运营活动平台迫在眉睫。该平台可以把业务前端页面的元素固化为楼层模板，并且将后台素材接口功能封装成组件，使各个业务方在搭建页面时可以像乐高玩具的零件一样，使用拖曳的方式快速简单地组装。

页面组装的可视化界面如图3-6所示，共包含四个部分。

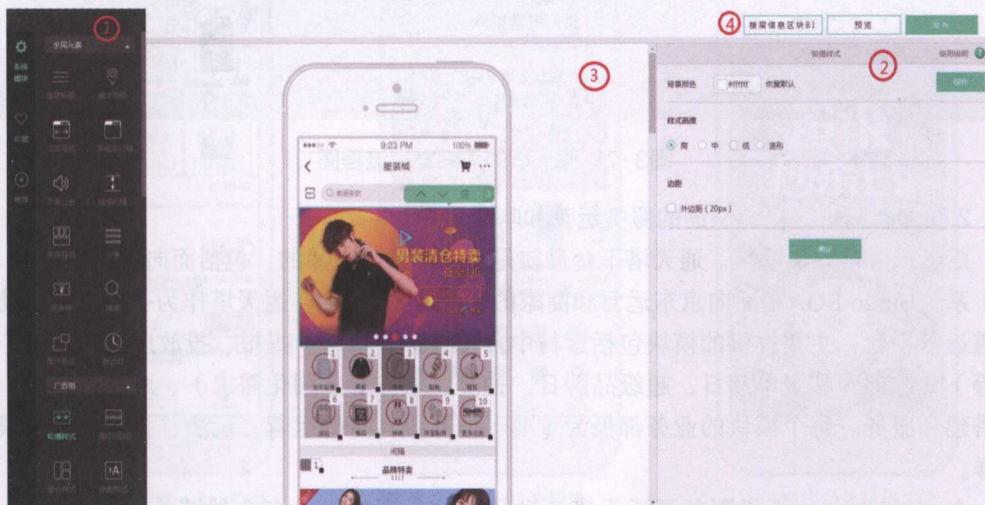


图3-6 页面组装的可视化界面

◆ **全平台自适应。**通天塔自诞生之初就秉持着“WriteOnce, RunAnywhere”理念，即一次编辑，全平台展示。运营进入通天塔可视化后台CMS系统，进入运营页面搭建好楼层并录入素材后，只需一键单击发布按钮，楼层配置数据就会保存到数据库。全平台包括原生客户端iOS、Android、M站、微信手Q，都将在1min内同步自适应展示最新搭建的通天塔页面，大大简化了复杂的上线发布审批流程。图3-7是通天塔的页面发布流程图。

在通天塔初期，前端只支持iOS、Android、M站三个平台，后续随着业务的发展，通天塔把页面搭建能力也同步给微信和手Q平台。

◆ **千人千面。**随着通天塔平台模板样式日益丰富，运营团队对通天塔提出了更高的要求。运营已不再像通天塔起步阶段时候那样抱怨模板样式不够丰富，因为丰富多彩的模板样式已经能满足运营促销的日常需要，反而更关心用户点击、流量和订单转化率。因此，通天塔需要根据用户喜好挖掘用户潜在需求——BI，目前通天塔已经支持各种不同维度的BI，如商品、广告、标签、楼层、楼层定投、优惠券等。

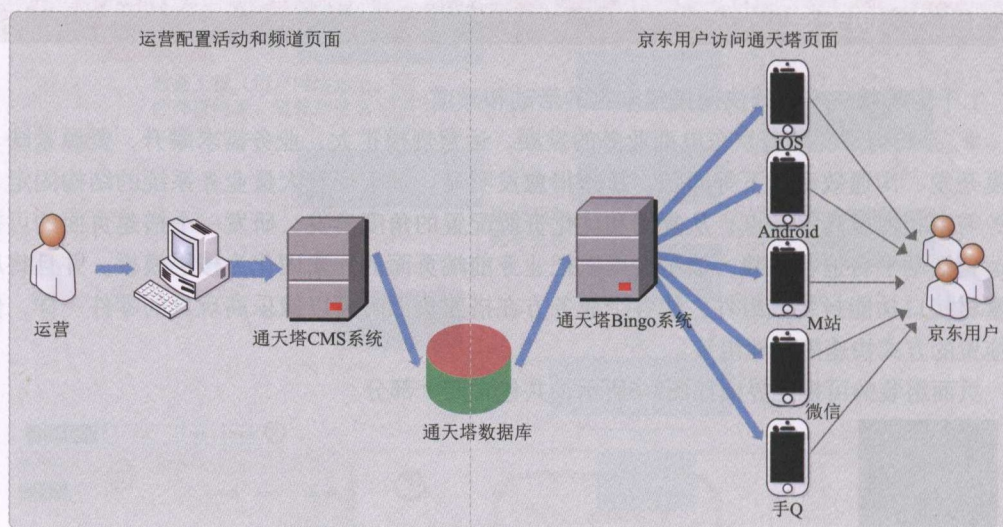


图3-7 通天塔页面发布流程图

2. 稳如磐石——稳健快速的服务运营和响应用户

系统拆分，模块解耦。通天塔平台最初是由两个系统支撑的，包括面向京东用户端（C端）系统Bingo SOA和面向京东运营和商家的CMS系统Babel。通天塔作为一个京东活动和频道运营平台，主要提供的模块包括素材中台（维护、运营、提报、投放广告、商品、素材等）、综促（超级单品日、超级品牌日、抽奖、领券等定制化需求）、可视化活动、频道搭建和服务。每个模块的业务都极为重要，但系统自身的运营、玩法、技术实现都很不一样。

◆ **素材中台。**不仅服务于通天塔活动和频道，还提供给多个外部业务方，如JShop等。所以对其稳定性、高并发能力、响应时间、系统开发迭代的敏捷性、独立性有着非常高的要求。采用只读服务。

◆ **综促。**平时流量不大，而在大促、超级品牌日等活动的开始时间点会突增，需要很强的瞬时流量处理能力，所以采用了异步化、限频等削峰机制，读写均等的服务。需求一般是根据运营促销活动来迭代的。

◆ **活动和频道可视化。**平时和大促期间都有很高的流量，承接了京东超市、京东生鲜等非常多的京东频道，对响应时间、吞吐能力、稳定性、高可用具有非常高的要求。采用只读服务。需求开发一般是按照京东App大版本来迭代的。

随着业务的发展，系统耦合的问题越来越严重，结合系统业务不同、流量情况不同、对系统要求不同、需求迭代不同等因素，于是开发人员决定把这三大块业务进行模块拆分，同时从技术实现角度思考，把相似的、和实时请求无直接关联的逻辑进行打包，拆分到新系统，并且把数据库基础查询逻辑封装到服务中。系统经过半年运行检验，得到如下提升：

①隔离异常、风险和波动，系统自身波动和问题不会传递给其他业务模块，避免某些

非核心流程影响核心流程。

②独立部署和维护，按照业务自身流量和性能需求独自横向扩展。

③独立项目管理，开发迭代更敏捷、更轻便。

④系统拆分带来数据库的垂直拆分，数据库性能更优。

3.1.4 秒杀系统

京东秒杀作为京东核心的促销频道之一，在京东商城中占有重要的地位，其通过超级秒杀日、超级秒杀、新品秒杀等限时促销活动，整合京东的优质资源，以提升用户体验和生活品质为核心，吸引了大量的用户积极参加。京东App秒杀前端界面如图3-8所示。



图3-8 京东App秒杀前端界面

海量的用户也给秒杀后台系统带来了巨大的挑战：

首先，京东秒杀作为整点促销，每天有10个场次，每个场次间隔2~6h不等。整点时刻，会有成千上万的用户同时涌入秒杀系统进行抢购，尤其是每年的618和超级秒杀日的海量流量会对系统和整个链路造成巨大的压力。如何解决切场前后的陡峭波峰流量，进行场次的平滑切换，在技术上难度较大。

其次，秒杀频道作为京东移动App首页和PC版首页的一级入口，又有微信手Q、开普勒项目等导流，每天服务亿万商家和用户，其重要性不言而喻，因此必须保证服务高可用。在实际场景中，各种极端条件下可能出现各种各样预料不到的问题，如何在复杂多变的极端条件下保证服务的高可用，是一个严峻的挑战。

自京东秒杀频道上线以来，京东秒杀稳定、可靠、流畅的用户体验得到了商家和用户的一致称赞，其背后少不了秒杀后台组开发夜以继日对系统的优化创新以及应急预案的完善梳理。本文将从海量服务、卓越性能、高效可用三方面探讨如何设计优化秒杀类系统，以及分享在实际问题中总结的宝贵经验。



1.海量服务——日常大规模高并发

为了应对日常的流量高峰，尤其是618、超级秒杀日等大促活动的海量流量，后台的架构必须可伸缩——只需要简单的新加机器就能应对出现的流量高峰。因此必须改变原有的一站式架构，一站式架构的最大问题是不同功能和模块之间的耦合太大，不易于扩展，难以维护。随着网站规模的扩大，对系统的扩容不可避免，而一站式的架构则非常不利于扩容，扩容之后的测试维护成本巨大。为了解决扩容难的问题，必须对不同功能模块和系统进行解耦，采用微服务架构，将整个系统拆分为秒杀、结算、风控、库存、推荐等各个微服务，各服务之间采用消息队列和RPC调用等方式传输数据。京东秒杀后台微服务架构如图3-9所示，多个服务之间低耦合。

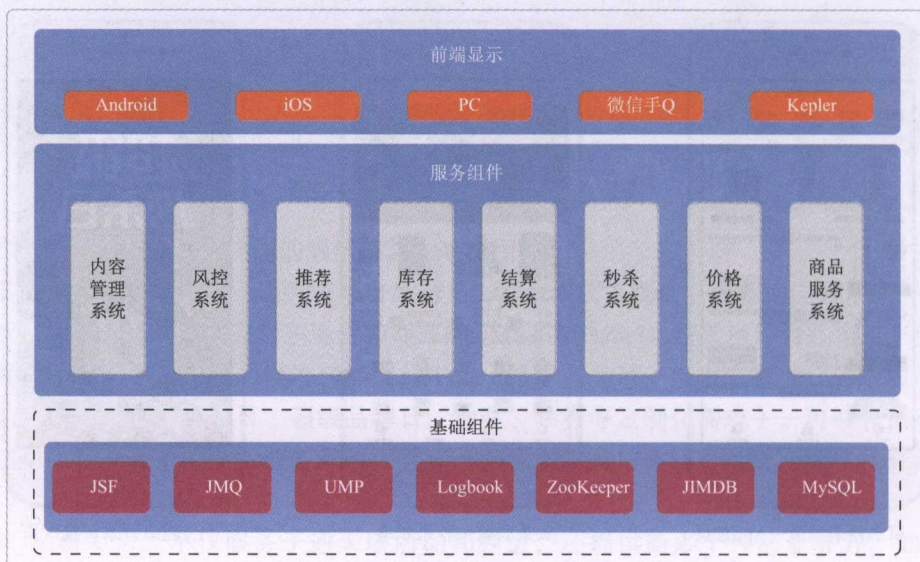


图3-9 京东秒杀后台微服务架构

采用微服务架构，每个服务安排一个团队维护，不同服务之间低耦合，有利于代码的开发和优化，相互之间独立工作，提高了开发效率。并且微服务的架构也更有利于进行水平扩展，按需扩容可只对需要的系统进行扩容，一方面有利于快速扩容，另一方面也降低了扩容对整体系统带来的潜在风险。

随着系统的不断扩大、机器的逐渐增多，数据库将会逐渐成为性能的瓶颈，直接读写数据库远远不能满足秒杀场景的吞吐量需求。常见的做法是采用Redis等基于内存的Key-Value数据库作为缓存，减轻对DB的压力。

2.卓越性能——瞬时流量涌入低延迟

对于秒杀场景最大的挑战是某一时刻流量的突然增大，即所谓的尖峰流量。在秒杀中经常会有当前场次结束之后，切换到下一场次的场景，切场前后往往是流量最大的时刻，解决该问题的一种重要思想是把瞬间的大流量变成一段时间的平缓流量。现有系统秒杀场次的切换基本都依赖于用户主动刷新，在切场前后用户会频繁地刷新页面，大量用户同

时向服务器请求数据，会产生百倍以上尖峰流量，瞬间加重了服务器、网络 and 数据库的压力。在京东秒杀系统中，开发人员采用切场前预加载的方式来减轻切场前后大流量对系统的压力，如图3-10所示。

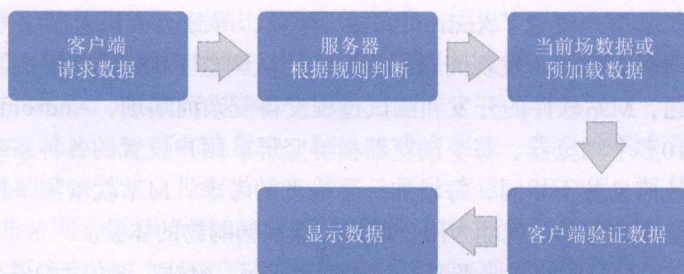


图3-10 切场预加载流程

在预加载时间段，客户端会向服务器请求数据，服务器根据一定规则判断下发当前场次数据或下一场次预加载数据。如果客户端预加载成功下一场次的的数据，则在场次到点切换之后不再向服务器发送请求，直接使用本地预加载数据。通过这种预加载的方式将整点切场的部分流量分摊到预加载时间段，大大削弱了切场整点的尖峰流量。

3. 高效可用——全年稳定无宕机

作为高速发展的业务，京东秒杀频道一直保持高效快速的迭代和升级，如何在迭代频繁的情况下保持线上业务服务的持续稳定可用，是海量在线服务面临的一大挑战。在京东秒杀频道内部，开发人员总结为三点：立体式的系统和业务监控、完善清晰的降级路径、严密科学的容错兜底逻辑。

3.1.5 M站

1. 流量万能承接者，从战战兢兢到稳健可控

◆ **M站的出现。**M站系统在京东存在了很长的时间，从2011年京东进入移动端开始的Wap版本开始，历经了非常多的变更，也遗留下了非常多的功能。最初M站的域名是wap.360buy.com，当时的名称是“标准版”，现在此版本已经完全下线，只零星有搜索引擎的SEO流量通过此域名进入京东，之后所有系统的变化和扩展都由这个域名开始。现在M站点的三级域名是m.jd.com，主要应用在Android和iOS手机上，便捷地服务于移动端客户。

◆ **服务的客户群。**作为京东在移动时代开始的主站，和京东的其他系统有非常多的耦合，M站的现有系统肩负了转化用户平台和服务于用户购买流程的诸多需求。

2. 全面的思考，在手机上有多少维度的可能

M站为了使每一位客户都得到较好的浏览体验，团队要兼容足够多的屏幕大小、足够多的手机型号、足够多的浏览器、足够多的用户设置。随着每年每个手机品牌厂商的市场占有率不同，团队还要进一步调整测试的范围。M站每个季度都要针对用户的手机浏览器状况进行数据分析。

但即使经过充分测试，团队也无法覆盖所有的用户，所以灰度发布和观察用户反馈就成为找到问题的方法。比如在2016年618前，有很多用户反馈自己的京东当中订单无法展示，团队做了多轮的覆盖测试仍然无法复现。最终依靠用户的一张截图，团队猜想到有可



能是用户设置了Android的特大字体，导致页面设置的字号被放大，挤占了订单展示的空间所引起的。发现问题原因后，团队调整了CSS样式解决了问题。由于有此类兼容性的问题，M站软件的开发和测试过程变得复杂而烦琐，Android和iOS两个操作系统、每个系统10多个浏览器、每个浏览器横屏竖屏、用户设置的各种系统字体变化、新老用户订单、地址信息各不相同，每增加一个维度的考虑，M站就增加一份开发测试成本，但每增加一个维度的考虑，就能为用户增加一点顺畅购物的体验。

M站的流量来源复杂，去向有App、M站、微信手Q各个目标场景。针对不同的渠道环境，可以做不同的处理判断，因此团队在618的准备期间，构建了渠道信息的收集、存储、配置、变化的流量基础设施，如图3-11所示。围绕M站的站点页面，团队构建了一套流量控制分发策略，不同渠道来源的用户，可以被精细地分发到更适合他们的平台和更精准的推荐商品，逐渐做到用户流量快速落地、精确分发产生最大转化的效果。

每年的618对于京东的技术人员都是一次巨大的考验，考验着团队的编码质量，也考验着团队应对突发事件的应急能力。M站作为一种轻便的访问模式逐渐被用户所认识，团队的系统也逐渐走向稳定，无论技术的演进朝着什么样的方向发展，无论技术与对抗环境有多么复杂，相信通过团队一次一次的大促间迭代的改进，使用手机浏览器访问京东的客户将持续得到更好的体验！

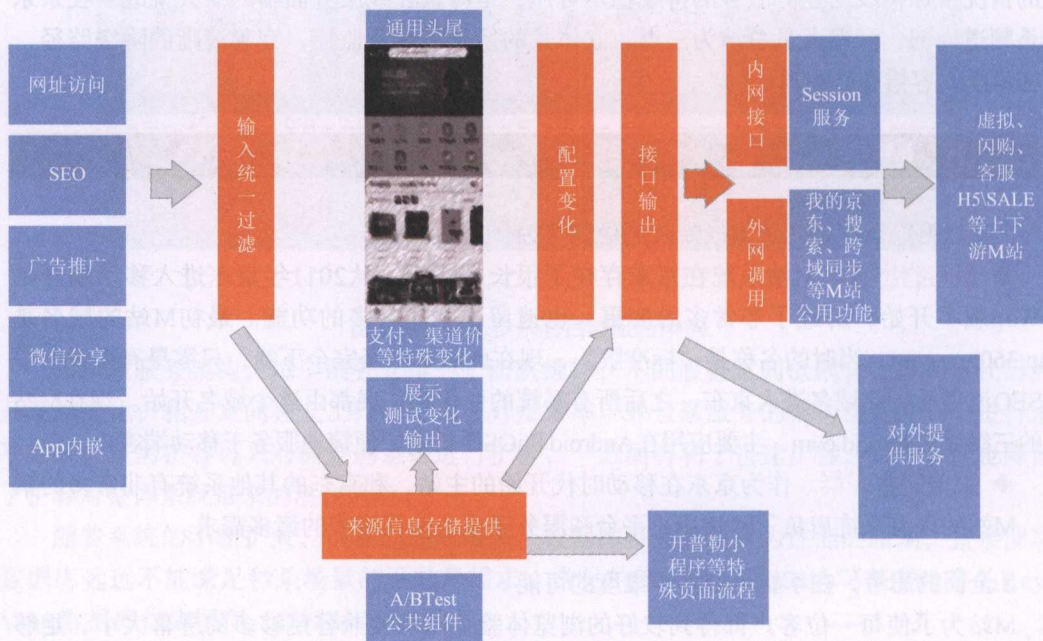


图3-11 流量基础设施

3.1.6 核心流程

1. 精细化流控——驯服海量访问

俗话说“能用机器解决的问题都不是问题”，但是再多的机器也架不住App端3年时间

百倍单量的增长。这给App交易主流程后端服务提出了更高的要求，用更少的机器扛更多的流量，如果这么大的流量瞬间透传到底层基础服务，一旦基础服务被击穿，后果不堪想象。因此前端需要建设一道安全的城墙，起到保护后端的作用。

首先分析流量组成，无论在平峰还是在高峰流量中，团队发现有很大一部分比例来源不是京东客户端，而是人为制造的非法请求，所以首先要清洗这部分流量。2015年，结算系统上了各种风控策略，效果非常明显且误杀率低，初见成效并不断扩大到各个核心系统。随着京东品类的不断扩充、渠道下沉、加上用户体验越来越好，在618时即便扛住所有正常流量也需要非常庞大的机器量，因此团队开始了精细化流控系统建设。

明确目标。精细化流控主要有如下两个目标：

- ◆ 提升服务器利用率，拉平峰值的流量，让用户的流量通过排队方式变得更加分散。同时前端增加排队的交互提示，让用户心里有一定预期，提升用户体验。
- ◆ 降级和流控策略更加细化，按照CPU利用率、负载、QPS、用户等级、地域、排队次数等，增加多维度的一些权重控制。

数据采集。在业务服务器中部署数据采集进程，该进程以秒为单位，采集CPU使用率、负载等性能指标，上报数据至流控服务系统共享内存中，供流控策略使用。图3-12所示为采集到的CPU使用率秒级数据分布。

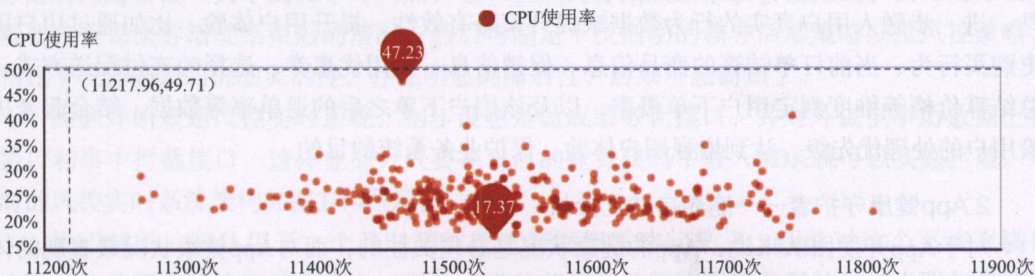


图3-12 精细化流控系统CPU使用率秒级数据分布

请求流量经过流控系统时，系统会将本次请求的用户等级、地域、请求接口、IDC等基础数据提取并上报至实时计算服务，实时计算服务将这些数据处理为秒级的请求分布报表，作为策略决策的基础数据。

流控策略。使用Nginx+Openresty将流控策略功能从业务服务前置到了前端Nginx模块中，作为独立的流控服务。

- ◆ 在前端Nginx中接入风控系统，非法流量将在防刷限流服务中被直接拦截。
- ◆ 对于超出业务服务处理能力的正常流量，综合考虑业务服务集群性能指标，如CPU使用率、负载、QPS等，把它们加到流控策略，指标值由上述数据采集系统收集。当这些性能指标达到一定阈值后，由于这些流量不是异常流量，不能直接生硬地拦截，所以流控服务与客户端建立排队策略，将这部分流量的请求结果直接设置为排队消息。客户端接收到排队消息后，将弹出排队提示，排队时间由服务端下发，排队时间达到后重新请求获取正常的业务数据。

◆ 进行入口检查和接口调用频率检查策略。



以上各种策略增加了模拟训练模式，在不影响线上流量的基础上分析策略效果、不断修正策略配置。

在实现流控能力的同时，将现有的流控服务产品化并对外输出能力，开发了服务管理平台，使所有业务系统能够快速具备防刷限流能力，系统架构如图3-13所示。



图3-13 精细化流控系统架构图

精细化流控的未来。精细化流控平台还需要真正地拥抱人工智能，使用深度学习工具，进一步融入用户真实的行为数据来提升策略有效性，提升用户体验。比如通过用户历史购买行为、当前订单结算的商品信息、促销信息、可用优惠券、选择的支付配送方式、总结算价格等维度判定用户下单概率，以及该用户下单之后的退单率等数据，结合起来决策用户的处理优先级，达到增强用户体验、保护业务系统的目的。

2.App健康守护者——崩溃自动化分析

对于App开发团队来说，App的健康状况是必须关注的，而对App健康状况最直观的体现就是崩溃率。通过对已经发出的App的崩溃率进行分析，可以知道这一版的App是否存在会导致大规模崩溃的未测试到的问题，以便及时补救。尤其是在618期间，App的更新幅度很大，每天还有各种配置化的运营活动上线，对App的健壮性挑战超乎以往。这时一个高效的崩溃收集平台就显得十分重要了。

目前大多数主流崩溃收集平台功能都很完善，团队最初的崩溃收集也和大多数平台的功能类似。但在上线一段时间后，团队发现了其中的一些不足之处，以及分析问题、定位问题时的痛点：

- ◆ 在收集信息后主要以崩溃类型来归纳统计，但往往某一崩溃类型下可能是不同业务模块所产生的崩溃问题。研发人员在定位时通常需要手动筛选日志，分析完日志后才能定位到具体的问题模块，但面对每天大量的崩溃日志，效率很低。同时，从系统监控上也无法直观地体现出真正出现问题的模块。

- ◆ 对于一些从崩溃信息无法定位的疑难问题，由于日志中寄存器信息有限，也没有系统事件的记录、原生页面和H5页面的访问路径，非常不利于问题的复现和定位。

- ◆ 通过逆向应用，以动态注入调试应用导致的崩溃，无法第一时间发现。

- ◆ 由于内存警告导致的崩溃问题不能捕获，无法对线上发生的该类问题进行监控。

当然，问题越多，痛点越多，也就越坚定了团队要从根本上解决这类问题的决心。团队知道，要提高用户体验，提升人员效率，首先要做的就是结合实际情况进行大胆创新。因此，团队升级了底层SDK，增加了内存警告崩溃的捕获。同时，完全自主地实现了一套崩溃自动化分析系统。该系统从日志分析、问题模块统计归类、日志展示、筛选维度以及监控上都做了更深层次的优化和探索。

通过多次大促期间的大量数据的检验，在系统功能性、分析结果准确性、定位问题及时性、监控精确性上都有了较大的提升，有效地提高了人员效率，整体崩溃率比历史同期也有明显的下降。

3. 无线业务安全

譬如说优惠券的安全，原先的处理方法是从日常的领券中甄别出一些机器刷票；分析申请的每个批次的优惠券的使用情况，多少券通过机器刷票被领走了，哪些品类的券经常受到机器刷票；然后再反馈给申请券活动的业务部门，帮助他们在后续的券活动中制定领券策略。目的是提高领券门槛，将机器刷票行为拒之门外。

这种类型的解决方案一直贯穿在无线各部门的业务安全防护措施中，但是随着机器刷票越来越精明，技术越来越高超，团队的监控策略也要相应跟着改变。京东开始发放优惠券的时间通常是一天中的几个时间点，基本上热门券刚放出来就会被抢光。当时的方案是需要分析每次券活动结束后情况，然后再制定下次活动的领券活动策略以及风控策略，相比于机器刷票的机动灵活性，存在明显的滞后性，后果可想而知了。

团队开始规划风控实时系统，初步设想是做成服务化接口，并对外提供事后数据上报接口和事中拦截接口，这样业务方只要在自己的业务代码中嵌入SDK就可以实施拦截。当时的风控实时系统架构如图3-14所示。

说干就干，团队用了一个月时间就把这套框架搭建起来了，团队对这个系统充满信心，给它取了个很响亮的名字——天神，意味着该系统能够看清凡间的一切光明与黑暗，保正义，惩罚邪恶。天神系统包括了核心的数据聚合及分发模块、流式计算模块、规则引擎模块三大功能模块，它们的作用分别如下：

◆ **数据聚合及分发模块。**它的作用是解析各个业务方传来的数据，并且可能在处理过程中需要聚合一些调用其他第三方接口回传回来的数据，在转换为统一的数据格式后，落入DB或者分发到每个业务的流式计算模块中。DB中的数据可供数据分析人员进行离线分析，开发可实施的风控策略并进行离线验证。

◆ **流式计算模块。**它的作用是接收数据聚合和分发模块传入的统一格式的业务请求数据，实时地实现各种复杂的风控策略，并且将风控的结果写入缓存中，供规则引擎模块查询。

◆ **规则引擎模块。**它的作用是为每个业务分配一个固定的业务标志AppID，然后通过传入的请求携带的AppID区分业务，进而明确需要查询的缓存，这个缓存中的数据恰恰是流式计算模块为该业务计算出的风控结果。按照这种方法，其实是把风控规则的实现异步化了，提升了业务方调用拦截接口的性能。

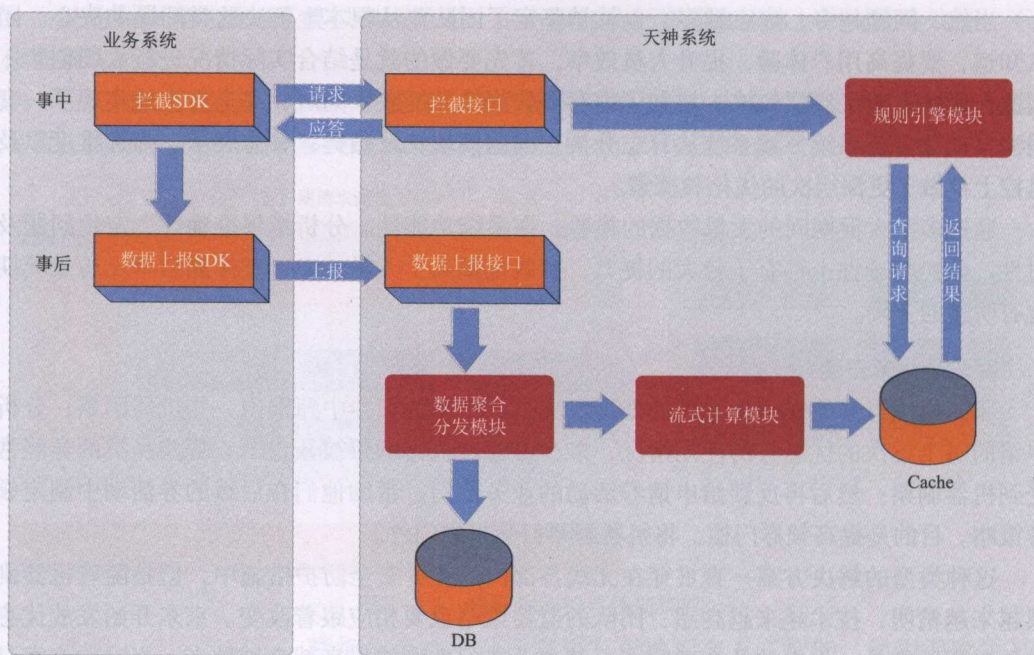


图3-14 原始风控实时系统架构

团队在反思技术架构的同时也在反思风控策略的问题，做数据分析的同事制定了两套策略，一套是在线上跑的实时干预策略，另一套是离线跑的监控策略。监控策略的一个作用是验证线上策略的准确率以及覆盖率，另外有些刚接入的业务暂时还没有实时干预的需求，就先上一个监控策略进行监控。实时干预策略有一定的规律性，但是监控策略就复杂多了，团队就在想这些策略是不是能够模板化呢？

按照这个思路，团队首先想到的是将原先线上的所有策略进行抽象和归纳，技术上先形成一套配置模板，所有的策略都可以通过这套模板来配置实现，这样需要改动的模块包括规则引擎模块和流式计算模块。

团队做的最重要的一件事就是天神安全网站，天神系统会把处理结果全部上报到这个网站，在网站上团队可以看见所有接入业务的实时拦截情况，以及对拦截原因的分析，还有对具体业务的监控策略的结果曲线。团队后来又把线上策略的配置界面化了，可以让数据分析人员在网站上直接配置线上风控策略，实时下发，实时生效。总之，有了这个网站，一切变得简单了，目前的架构演变为如图3-15所示。

其中的规则仓库储存着每个业务对应的风控规则，这些规则由天神管理平台下发和维护。

2017年的618，京东向社会递交了一个惊人的数字，1199亿元！这场席卷全中国的大促对国人来说是场福利，而对京东技术人来说则是个艰巨的任务，对天神系统来说更是场大考。包括下单、商品详情页、支付、购物车、搜索、统一登录、京东通天塔在内的众多无线核心业务已陆续接入天神系统，天神系统在为众多业务保驾护航的同时，也在检测团队自身的技术实力。100万次/s的峰值QPS，99.99%的峰值可用率，以及1~2ms级别的请求时延，这是天神系统向接入业务方交出的数字。

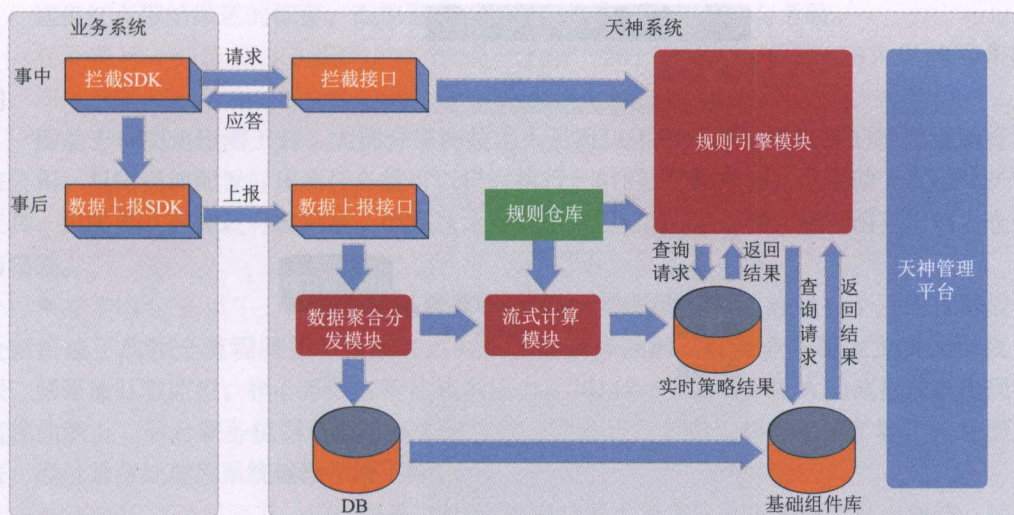


图3-15 改进后的实时风控系统架构

2017年的618过去了，但是团队未来的路还很长，安全的道路上没有终点，只要“坏人”存在一天，团队就会始终坚守。



Civic：平台产品研发部技术服务负责人，负责前台产品质量方面工作。

3.2 渠道引流核武器——微信手Q购物

京东与腾讯的联姻诞生了京东微信购物、手机QQ购物（以下简称微信手Q）渠道。2014年5月27日，微信购物入口第一次揭开面纱，成为京东的一个重要购物入口。作为社交电商的领头羊，微信手Q购物入口持续创造出新颖多变的玩法，不断刷新用户体验的同时，也成为京东流量引入和拉新的功臣。

3.2.1 业务支撑修炼之道

1. 红点——流量冲击带来的惊喜与考验

红点惊喜——微信手Q入口的618秘密武器。2014年5月27日，微信购物入口第一次投放了针对618活动的红点，如图3-16所示。吸引用户的关注和点击，它是渠道购物引流的秘密武器，在每年的618期间都会为微信手Q购物渠道带来可观的流量。经过3年的沉淀，微信手Q购物不仅在业务流量和成交量上不断增长，在后端的技术积累上也不断夯实，从而更加坦然地面对618的来临。

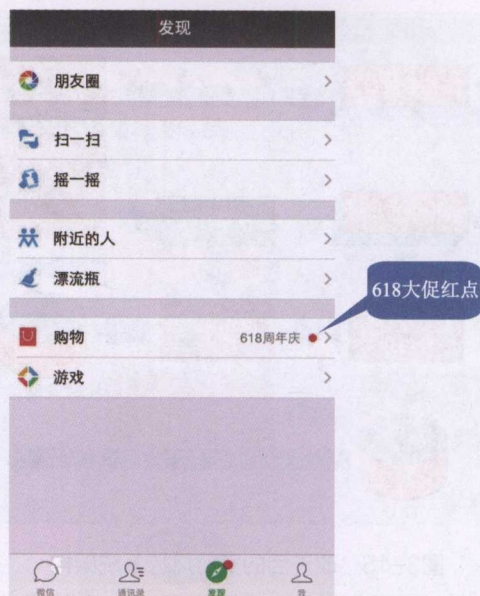


图3-16 微信购物渠道红点展示

2017年6月18日11:00，红点投放的时间越来越近，购物大入口首页的监控曲线开始陡升——流量来了。此次微信红点的投放时间在18日中午，流量如火箭发射般瞬间飙升了十几倍，如图3-17所示。

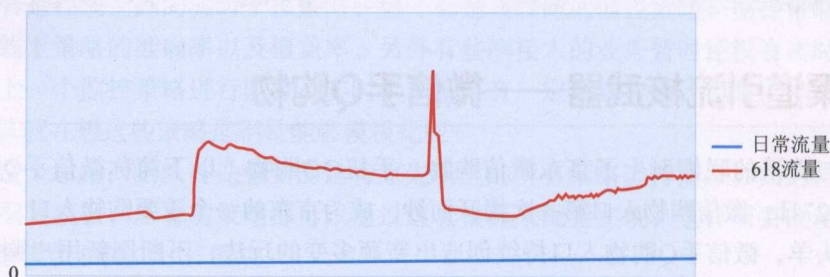


图3-17 2017年618流量曲线

从2014年的618到2017年的618，红点带来的流量不仅仅为渠道吸引了新用户，也带来了成交量的增长，图3-18和图3-19所示分别为历年618UV和历年618PV。

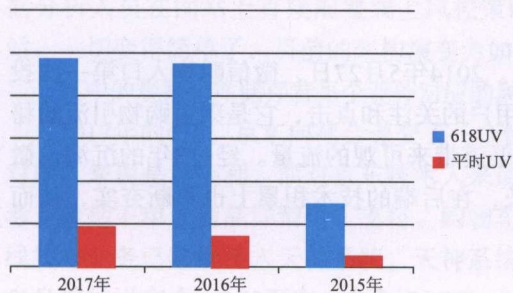


图3-18 历年618UV

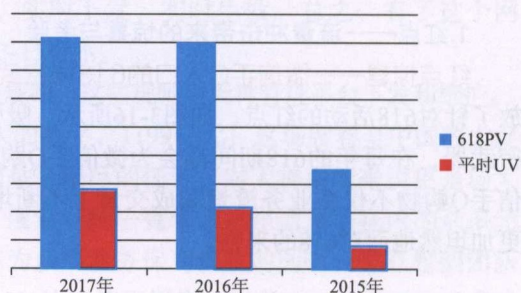


图3-19 历年618PV

这是红点带给渠道的惊喜，而惊喜背后则是对技术巨大的挑战与考验。

红点考验——流量冲击应对的技术演进之路。2017年4月，备战618大促的号角吹响。

得益于自动化压测工具，大部分压测技术人员可以不用再在月黑风高的凌晨艰难打车来公司，只需提前配置好压测任务就可以定时执行。618前夕的扩容，也借助了自动化扩容工具，只需要几分钟就可以完成。而在这之前对618红点的应对过程，相对而言却略显仓促与狼狈。

◆ **2014年：初试牛刀。**2014年是微信手Q购物入口的诞生年，经过一个月的攻坚战，全新搭建了购物全流程服务。投放红点时，流量来势汹汹，红点业务研发负责人彻夜未眠，紧张地盯着监控，担心系统扛不住流量压力。2014年5月27日，巨大的流量请求出现在监控曲线上。后台服务机器负载过高告警，红点业务研发负责人临时手工扩容，忙乱而紧张，经过紧急处理后系统逐渐平缓下降。

◆ **2015年：静态容灾。**后台服务宕机问题在618的大流量洗礼下是很容易出现的，接着就会出现页面空窗，对用户体验造成巨大的影响，进而可能是业务成交损失几百万元。

所以，静态容灾就应运而生了。服务正常运行时把传到前端的数据存起来，并不断更新，当服务出现问题时，这些数据就起作用了。虽然是历史静态页面数据，但是除少量的细节外，和正常访问的页面并无太大区别，成功避免了前端空窗这种尴尬局面的发生。

◆ **2016年：例行容灾演习与自动化压测。**如果说前两年的618是人工备战，那么2016年的618就是自动化备战了。对于微信手Q渠道来说，618期间的流量可能是平时的几十倍。对于这种挑战，后台曝光服务平时会采取自动化压测来了解机器的性能。压测方案是采用请求重放的方式来对服务进行压测的，开发人员先在线上环境采集真实的流量数据，保存到压测平台上，然后使用TCPCopy等工具将请求重放到指定的服务机器上。另一方面，为了能更加从容地应对618中可能出现的各种服务异常，团队开始了双周例行容灾练习，通过不断的日常演练来提升异常处理的熟练度。

◆ **2017年：高效备战。**

① **流量预估。**在618前的一个月，开发人员会根据上一年618流量和平时的流量估算出各个服务当年的流量。同时会梳理每个业务服务当前的机器数。

② **压测。**有了自动化的压测工具，会根据预估的流量进行压测，并评估当前的服务性能。评估当前的机器是否足以保证大流量下服务的可用率，以及为了应对红点的流量高峰还需要多少台机器。

③ **自动扩容。**有了机器，接下来就需要对机器资源还不足的服务进行扩容。相比于2014年618一个负责人紧张而又忙碌地对80多台机器一步步地手工扩容，2017年的扩容显得格外轻松。自动扩容工具的研发，使得现在的扩容操作只需要5min即可解决，备战效率实现了质的飞跃。

④ **容灾降级。**如果说所有的压测扩容都是为了红点来临时能够平稳度过，那么应对流量冲击的最后一道防线就是降级。对于一个电商的后台服务来说，最主要的任务就是让用户能够通过浏览器看到商品，如果连这一点都有问题，那么服务已经是不可用的了。除商



品展示等主要功能外，为了便于分析数据，服务接入了数据上报的功能；为了优化效果，服务还接入了BI排序，根据用户的购买和浏览历史等信息来展示用户最喜欢的商品。而这些主要功能之外的都属于可降级的逻辑步骤。所以，在618期间，当服务难以承受住巨大的流量压力时，可以把这些旁路逻辑去掉，或者按百分比使用，以减轻服务的压力，保证主要逻辑能正常运行，使影响用户体验的效果降到最小。

2.SNS活动平台的实践和演进

初探。2014年，是京东微信手Q事业部第一次经历618，部门承接的第一个微信环境SNS活动是刮红包。由于新鲜的H5体验和微信手Q强大的社交属性，加上前所未有的优惠券力度，使得活动达到了病毒式的传播效果，参与人数迅速暴增。

发展。为了让运营人员可以自由配置，减少开发人员介入，微信手Q渠道开发了营销活动管理系统。运营人员在系统中可以自主配置奖品的类型、发放数量、抽奖概率、用户中奖上限等信息，而开发人员只需要调用公共组件方法即可。

促销活动的系统架构如图3-20所示，其中规则池模块负责限制领取门槛，例如必须是实名账号或关注京东大账号等，核心抽奖模块负责根据活动规则配置去概率抽奖或兑奖，然后修改数据库扣减奖品数量，发奖网关则负责异步给用户发奖。

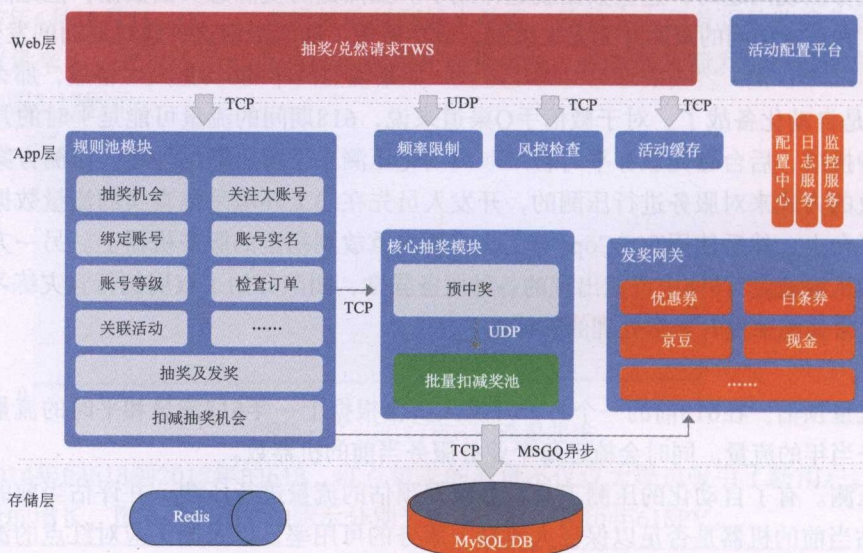


图3-20 促销活动的系统架构图

因为每个活动的门槛不同但又有交集，所以团队创建了规则池模块，每个规则对应一个动态加载模块，新增规则时只需新增动态加载，降低了代码修改风险，提高了开发效率。而规则又分为前置规则和后置规则，像拦截器一样，在抽奖前后按照编号顺序对用户数据进行判断和修改。

有了营销活动管理系统后，开发效率大大提高，加上各种接口的组件化，使得从最开始的一个月开发一个活动，到现在一周就能开发五六个活动，效率提升了几十倍，而且系统也更稳定、更灵活了。

完善

◆ **自动化运营**。由于发券的需求越来越多，但开发资源有限，因此开发人员将很多效果好的活动做了改造。各个类目的运营人员可以配置各自个性化的皮肤和奖池信息，并设置好活动的有效期。而前端页面则通过引用运营人员配置的页面片（SSI），读取配置信息，即时发布上线。

◆ **多方合作**。因为SNS活动的火爆，越来越多的外部合作方找到微信手Q购物渠道，希望能支持发放京东优惠券。针对不同场景团队提供了不同的接口和固化页面，以满足不同合作方的需求。

◆ **实时监控**。针对在多平台配置的活动，团队开发了啄木鸟系统，用来校验各个系统配置的正确性和完整性，只要发现错误即发出告警，提示负责人修改。

◆ **移动运维**。当下班时间有系统告警或者用户投诉时，为了登录VPN，就需要快速找到计算机和网络，然而并不是每次都能如愿。为了保证能快速而有效地发现问题原因，研发团队将平时用到的工具进行了移动化处理：在京东内部使用的App中嵌入了用户信息查询、发奖情况查询和监控视图查看等功能，保证无论何时何地都能快速发现和解决问题，使得运维效率得以大幅提升。

3. 一次基于微信关系链的内容化社交化电商之旅——购物圈

购物圈关系链。购物圈关系链由两部分组成：基于应用的微信朋友圈关系链和购物圈自身关注关系链。

◆ **基于应用的朋友圈关系链**。基于应用的朋友圈关系链是微信朋友圈关系链的一个子集。

在购物圈场景下，用户登录购物圈，每隔24h获取微信是否授权信息，在授权状态下通过微信网关尝试上报微信行为，并且异步通过微信网关向微信拉取一次关系链并保存。

◆ **购物圈自身关注关系链**。这部分关系链是用户在购物圈体内，主动关注其他用户而生成的。例如，用户对某个明星或达人经常分享的文章感兴趣，他（她）可以关注对方，这样对方每次分享的晒单，都会出现在他（她）的“好友动态”里。

购物圈系统演进。购物圈作为一个多维度的服务平台，同时也管理着非常庞大的数据和数据之间的关系。目前管理的数据主要分为：Feed（晒单）、用户、标签、点赞、评论、消息等，数据又有各种维度的聚合，如图3-21所示。

购物圈容灾

◆ 容灾背景

①购物圈和内容自助平台的前端数据展示严重依赖Redis，而前端页面不能因为Redis宕机就浏览不到内容，可用性是首位。

②有些场景对数据实时性要求没那么高，可以直接提取JDCloud（京东云）上的数据，减轻Web服务器接口压力。

③对于长文章的富文本信息，可以直接储存在JDCloud上，后端不用考虑存储结构，推送到TDCloud即可，同时也减轻了Redis和DB的存储压力。

鉴于以上背景，研发团队开发了购物圈的容灾系统。

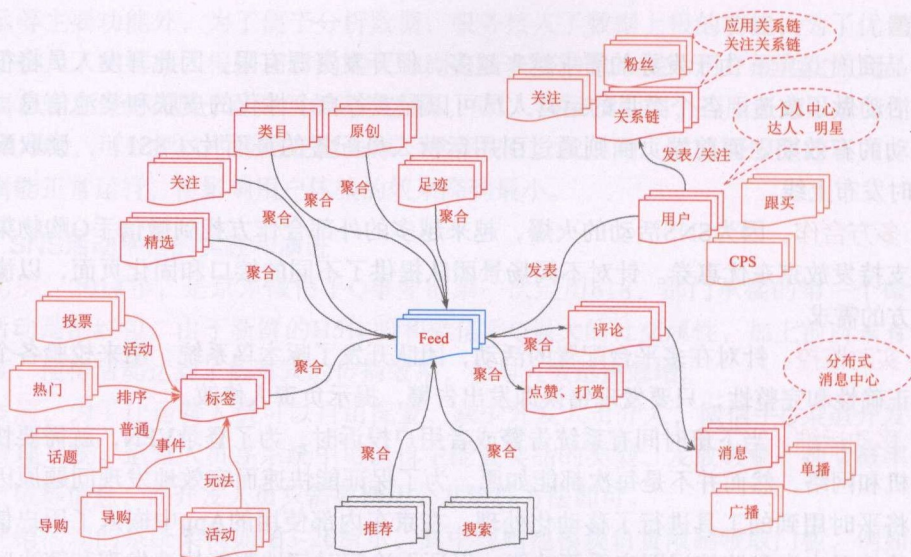


图3-21 购物圈数据各种维度的聚合

◆ **容灾系统简介。**容灾系统主要考虑对于静态数据文件的处理，容灾静态数据文件生成流程如图3-22所示。

业务方通过约定的协议向JDCloud推送中心发送ToyQ消息。JDCloud推送中心的Daemon模块用来接收业务方的ToyQ消息，并生成容灾文件，同时调用UpLoadAo接口推送文件到JDCloud。业务方不需要关心文件推送是否成功，由JDCloud推送中心Daemon来保障推送成功。该系统是建立在ToyQ消息100%可用基础之上的，当业务方发现ToyQ发送失败时，需要业务方重新发送。

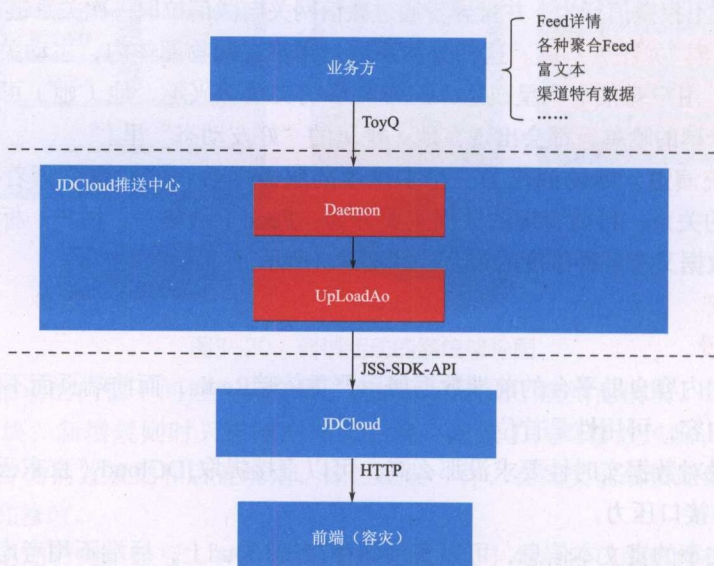


图3-22 容灾静态数据文件生成流程

3.2.2 技术运营发展之路

1. 立体监控

概述。在大型分布式系统中，无一例外地都遇到如何监控数量庞大的机器设备群的问题。在公司内部，有众多优秀的监控系统，它们在自己的专业领域发挥着极大的作用。然而，还是存在着部分监控缺失，或者监控点过多过散，无法全局把控的问题。针对此痛点，团队启动了Athena雅典娜体系项目，希望“智慧女神”能给我们指引，搭建了一套全方位、智能化的监控体系。

首先是全方位，从宏观到微观的粒度划分，团队规划了红绿灯系统、模块间调用系统、执行漏斗分析系统，实现粒度全覆盖；按照场景的不同划分，团队规划了前端测速系统、错误码上报系统、前端统计系统，结合现有的各种后台监控平台、CDN监控平台，实现了从前端到后台的场景全覆盖。

然后是智能化，首先针对众多监控数据进行了优化和治理，并在数据之上搭建一套功能强大的故障识别引擎，并借助大数据挖掘和人工智能，实现风险预警以及故障识别引擎优化。识别故障之后，除常规通知外，引擎还会将告警详情通过消息队列广播出来，由故障处理服务进行故障分析，并尝试自动恢复。

监控方阵介绍。整个雅典娜体系的全局架构如图3-23所示，大体可分为数据采集、故障识别、故障处理三大模块。

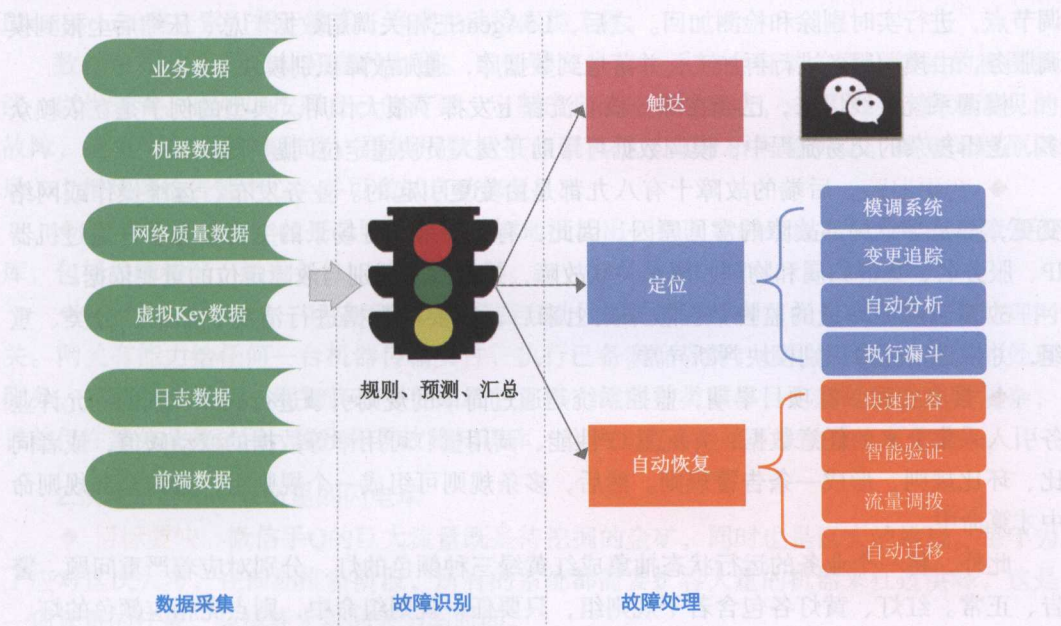


图3-23 雅典娜体系的全局架构

数据采集。在项目早期，团队的重点目标是梳理补全数据的上报渠道。包括整合公司现有的UMP系统，也包括自建的红绿灯渠道、虚拟Key渠道（允许业务自由定义上报），更包括前端上报渠道。一些现有的渠道，如机器数据、网络数据、日志数据，也提供了转



换方法，变成更统一的结构化数据，方便后续步骤关联分析。

◆ **红绿灯渠道**。红绿灯渠道是业务数据的主流渠道。业务根据需要申请监控指标，调用雅典娜的API即可完成上报。为了提高性能，团队提出了一种正在申请专利的数据压缩、本地汇总、集中上报方案。

上报到数据中心后，为了达到更高性能，系统设置了缓存层，用于数据快速写入与跨机器数据合并。最终会将数据落地到数据库，并实时通知故障识别模块的规则引擎，告知数据已经准备完成。

红绿灯数据以分钟为粒度记录，得益于高效的采集渠道，故障识别模块可以在10s内采集完数据并得出识别结论。

◆ **秒级渠道**。红绿灯的分钟粒度，加上10s延时，对于多数业务来说，已经足够实时了。但在极端情况下，若故障发生在分钟早期，最多可能会有 $60s+10s=70s$ 的延时，对于敏感业务来说是无法接受的。因此，团队实现了秒级渠道，满足这部分业务的特殊诉求。

秒级渠道的架构设计和红绿灯分钟渠道类似，采用了本地消息队列和秒级累积压缩。为了更实时，将上报的协议改成了UDP传输，鉴于内网的质量和数据包的大小，实时性和可靠性都比较有保证。2017年618期间，秒级渠道由于超高的实时性和非常友好的体验，得到了各个业务的广泛喜爱。

◆ **模调渠道**。模调数据采集模块，是集成于微信手Q渠道配置中心组件的负载均衡服务L5Agent中的。L5的职责是帮助调用方选择最适合的被调模块，并且检测出发生异常的被调节点，进行实时剔除和检测加回。之后，L5Agent把相关调用数据汇总、压缩后上报到模调服务，由模调服务进行再加工，并落地到数据库，通知故障识别模块。

模调系统上线以来，已经在部分核心流程上发挥了很大作用。典型的例子是在依赖众多、逻辑复杂的交易流程中，模调数据可帮助开发人员快速定位问题所在。

◆ **变更追踪**。后端的故障十有八九都是由变更引起的。业务发布、运维操作或网络变更，都是导致局部故障的常见原因。因此，有必要采集各场景的变更信息，并通过机器IP、服务名、集群归属和物理归属来关联故障，提供故障识别与故障定位的重要依据。

故障识别。海量的监控数据被采集上来后，需要对数据进行清洗、加工、分类、重组，并推送给故障识别模块判断异常。

◆ **规则引擎**。在项目早期，监控系统是通过简单的规则引擎进行故障识别的。允许业务引入采集上来的任意数据，并配置TP性能、调用量、可用率等数据的变动阈值，或者同比、环比规则，形成一条告警规则。然后，多条规则可组成一个规则组，组内全部规则命中才算命中。

此外，将一个业务的运行状态抽象成红黄绿三种颜色的灯，分别对应着严重问题、警告、正常。红灯、黄灯各包含若干规则组，只要任一规则组命中，则点亮相应颜色的灯。红灯优先判断，红灯亮时则不再判断黄灯规则组。

看似简单的规则、规则组，实际上覆盖了大部分的常见需求。有了这些规则组，可将众多零散的监控点有机地组织在一起，最终抽象成业务的整体运行状态。这节省了大量逐个看监控视图的时间，使得开发人员能更专注于业务开发。在持续的使用过程中，根据实

际需求优化了几版，增加四则运算、分时间段生效、假恢复等个性功能，使得规则引擎更加稳定和成熟。

◆ **宏观视图**。有了业务整体状态之后，根据业务流程，组织成一张壮观的业务状态树型图，整个核心流程的运行状态一目了然。一旦发生故障，相应业务的灯颜色随之变化，并发出告警声。

◆ **走向智能**。熟悉电商的读者可能会留意到，电商业务由于促销、运营等需要，会给系统带来一些有规律的尖刺。这些尖刺往往会洞穿相对严格的规则，造成一定的告警骚扰。那么，这些规律是怎样的呢？研发人员深度挖掘了历史的业务数据，尝试对业务尖刺进行预测，让规则引擎提前适配。对规律性较强的常规尖刺，预测准确率很高，而对于规划中的重要引流活动，面临的另一个问题是大促销流量高峰时的扩容，如何根据业务运行情况，得知现有集群的容量上限？又如何决策需要扩容的量呢？2017年初，微信手Q渠道启动了容量项目，目标是监控全部业务的容量情况，并提供扩容建议。

故障处理。发现故障之后，必须第一时间响应，在最短的时间内解决问题。在分布式系统中，大部分的故障都会被容错容灾系统屏蔽起来，不会对最终用户产生影响，但仍然需要尽快定位解决。可是，故障类型林林总总，如何快速定位并解决呢？答案是：定位工具、故障知识库、自动处理网关“三剑客”。

◆ **定位工具**。采集上来的数据中，通常蕴含着大量信息，指向故障的原因。但如何高效便捷地从海量数据中挖出想要的信息，则要分层整理汇总，对数据适当冗余，并存入ElasticSearch建立索引便于检索，形成主动检索类工具。

数据可视化也是定位问题的利器。当告警发生时，系统会自动关联出相关的视图曲线，以供查看。也可以按照自己的需要，定制出自己的视图集合，一览无余。针对常见的故障，业务可收听故障消息，再调用数据接口或其他工具分析问题所在。得知问题原因后，上报反馈给告警中心，并可尝试自动修复。

◆ **故障知识库**。定位工具阶段，会定期地沉淀出一些固定的处理方案，形成故障知识库，包括故障特征、触发条件、处理动作等。

◆ **自动处理网关**。一旦故障确认，需要自动处理的时候，即可提交指令给自动处理网关。网关有能力给任何一台机器传输文件，执行已备案命令等，也可调用业务提供的处理服务，实现服务切换、配置修改等更细致的工作。故障种类繁多，自动处理也非常复杂，目前仅在部分业务试用，逐步积累故障知识库。这块也将会是接下来的重点工作之一。

2.快速扩容——队伍里的闪电军

◆ **目标要快**。微信手Q的巨大流量既是待挖掘的金矿，同时也是巨大的挑战。每年为了应对这次大考，在前期准备阶段，所有的系统都需要扩容大量的机器来扛过洪峰，这是一项繁重的任务，会消耗大量的人力与时间。

必须有一个强大的快速扩容系统，来减轻繁重的扩容工作。在经历了微信手Q渠道的第一个618之后，这个目标就坚定地树立起来了，快速扩容系统应运而生，对应的流程与结构如图3-24所示。

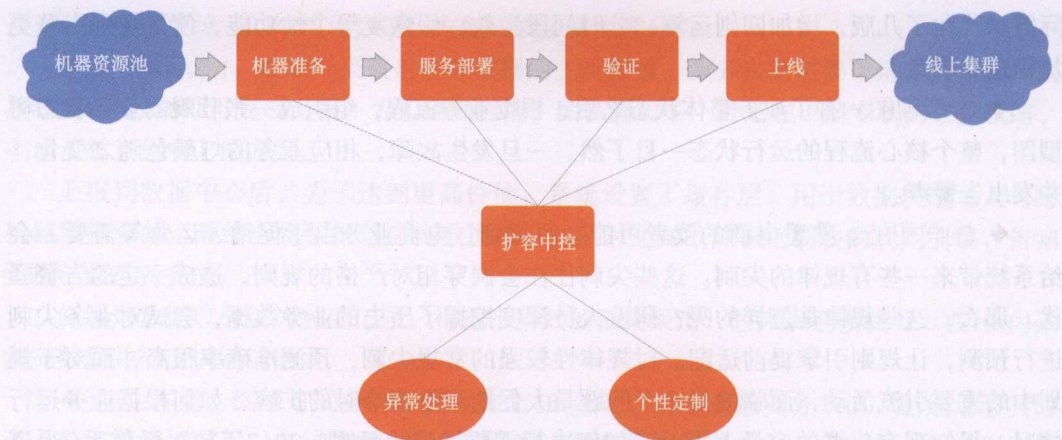


图3-24 快速扩容系统流程与结构

◆ **深入分析，寻找方向。**扩容过程的特点是流程长、环节多、每个环节步骤比较琐碎，而且不同的环节、步骤需要不同工作岗位角色的参与，所以是一个比较耗费时间、精力的过程。这样的一个过程需要提速，很自然的方向就是将它变成自动化的流水线执行过程，减少过程里的人工参与，这样才可以在释放人力的同时，提高执行的速度。

在这个方向的指导下，最终形成的方案是一个中控式、模板化的扩容流程系统。扩容中控系统在最高层负责流程的调控、执行；在扩容流程各环节的层面上，扩容中控系统提供模板化的能力，把每个环节里的各个操作步骤整合到模板里来，供高层的中控系统进行操控执行。在更下一层的步骤层面上，则需要把各种具体操作变成可以由机器执行的标准化接口。

◆ **扩容步骤标准化。**征程的第一步，是在步骤的层面上进行规范化、标准化，这便是一个大工程。之前扩容的各种步骤，是由不同岗位角色的同事分别执行的，使用的工具也是五花八门，有网页操作、各种格式的API，甚至有各种可执行程序工具等。只有把这些操作方式都规范成为标准化的接口，才能让中控系统流水线式地执行。

◆ **扩容环节模板化。**模板化是为了让需要按特定顺序执行的步骤集合，可以由机器程序重复地进行操作。这样一来就减少了人工的参与，不需要各种角色岗位都到位即可进行，保证了这个流程的速度和紧凑节奏，离“快”更近一步。

◆ **扩容流程的搭建。**扩容环节模板化完成后，扩容流程的模板化便是水到渠成了，因为各个环节间的顺序是相对固定的，因此在流程的层面中控系统只需要把它们按顺序联结起来即可，不过在这过程中仍然有两个方面需要小心应对。

一方面是流程中的异常处理。由于扩容各环节中步骤繁多，所以出现各种失败、错误的异常情况也较多。当出现异常情况时，中控系统需针对不同的异常进行问题处理、步骤重试，因此中控系统包含了一个可扩展的异常处理模块，可以灵活应对各种异常。

另一方面是个性化的需求。由于各个业务特点不同，扩容的过程中可能存在一些特有的步骤和操作。为了应对这种情况，快速扩容系统在上线环节之前插入了一个可选的个性化环节，各业务可以在这个环节配置自己特有的操作模板，实现自己的个性化需求。

◆ **功夫在日常。**搭好快速扩容的框架之后，大量功夫就在日常的磨炼里了。

快速扩容系统提供的是模板化的能力，发挥它的能力，一方面需要业务接入进来，将过去发布流程零散的工作——配置、编写到各个环节的模板里。这是一次比较琐碎的初始阵痛，但是为了具备快速扩容的能力，每一个业务都很好地完成了。

另一方面，业务又是在不断变化的，每当有变化的时候，就需要同步更新相应的模板，保持模板的最新状态。这样平时分散在不同时间段里的小块维护工作，最后汇聚成了在必要的时候“快”的能力。

◆ **苦修终成正果。**自从快速扩容系统搭建并接入线上使用之后，618等各种促销来临时所要进行的扩容，便不再是一件苦差。以往一个业务可能要花一整天才能完成几十、上百台机器的扩容，现在几分钟、十几分钟就能轻松快速地搞定。研发团队在人力得到大大释放的同时，也能更从容、稳定地应对各种流量的压力，分出更多的精力来保障系统运行的其他方方面面。

3.自动容灾——时刻保证业务高可用

微信手Q购物后台系统的服务都是按多SET部署，一个SET即一个单元，单个业务请求都在SET内完成。SET化部署既方便对业务服务进行评估、扩容和分流，又方便实现自动容灾的处理。

对于数据容灾，开发人员根据业务的重要程度分别采用了不同的容灾处理。对于一般服务的数据，采用一主两备跨机房的灾备模式；对于重要服务的数据，采用同城双活的灾备模式；对于核心服务的数据，则实现了异地双活。随着2017年广州机房的投入使用，核心服务将进一步扩展至异地多活。

4.质量保障——光与影的较量

在这场公司全员参与的618战役中，作为后端支持部门的质量管理团队可谓“海陆空”齐上阵，在为业务质量保驾护航的过程中，团队在前端页面的测试与监控、后台接口自动化测试、自动化压测，以及自动化安全扫描与一站式智能发布验证等方面做了系统性的创新和突破，并取得了很好的成绩。同时，在京东购物小程序方面，团队也做了大胆的测试尝试和创新，探索了新业务新模式下的测试方法。整个质量保障体系在日常工作及618实战中不断演变及完善，逐渐建设起一套适合自身业务的工作方法。

前端页面扫描与监控。常见的电商类业务场景中，用户接触最多的是运营活动、卖场页面、商品详情页等，前端页面的功能和体验在很大程度上决定了用户的去留，而这些业务都具有变化快速的特性，导致常规的UI层面自动化并不太适用。因此，团队希望从另一个角度来解决问题，既把用户体验和数据层面的监控作为切入点，在日常测试和线上监控过程中，开发并引入了前端页面性能扫描工具和前端页面功能监控扫描工具，从而在测试效率和监控效果方面得到了很大的提升。

◆ **前端页面性能扫描。**微信手Q业务部的前端页面均为H5页面，在性能测试方面，传统的PC或者插桩方案均存在一定的局限性。为获取到更加准确直观的性能数据，团队搭建了一套基于真机的实验室，开发了前端性能扫描工具，可以在测试中和上线前对页面进行



一键性能扫描。

前端性能扫描工具基于真机访问页面，获取页面首屏大小、页面加载速度、页面请求个数等数据，并将抓取的页面请求按照自定义规则进行分析检查。开发和测试同事只需将页面URL和投放场景录入工具中，工具就会自动进行扫描，并将扫描结果及存在的问题详情反馈到报告中，供开发和测试同事参考使用。

另外，该工具已集成到发布系统中，当开发人员进行页面发布操作时，发布系统会自动触发对该次发布的页面进行性能扫描。因此，即使是免测试的版本，开发人员也能及时收到相关性能检测报告，及时修复性能问题，确保上线质量。

◆ **前端页面功能监控**。618期间，上线的活动及卖场页面繁多，这些页面上的商品及资讯信息的正确性对用户来说尤为重要，例如，页面中的链接是否正确、商品数据是否为空、商品价格或库存是否显示正确等。原来这些信息的检查都完全由人工来完成，非常耗费人力和时间，所以，微信手Q购物渠道根据此情况开发了对应的监控工具，旨在通过监控扫描确保线上环境中活动及卖场页面数据的正确性，提升页面上线的工作效率。

后台接口自动化测试。通常情况下，在后台测试过程中，私有业务的后端服务使用的都是非标准通信协议，而非标准协议自动化测试所面临的核心问题之一就是协议分析。协议分析往往因开发语言本身特性不同而差异很大，目前最常用的两种策略为正向识别和逆向推导。

微信手Q购物渠道的后台服务主要采用C++语言进行编码，但C++语言不支持运行时动态获取这些协议关键信息。传统的后台接口测试工具，需要增量硬编码协议栈逻辑和接口调用代码，来支持新的协议和接口，协议适配和接口维护成本很高。

针对此情况，微信手Q购物渠道设计了一套以插件方式管理协议栈和接口解析逻辑的后台自动化测试系统。协议开发者按照既定规范，将协议栈和接口解析逻辑封装为插件（语言可选）、一次性录入，用户增量上传接口定义文件即可开始测试管理过程，极大地降低了协议接口接入成本及工具使用成本。目前团队已经上线了近200个用例，主要应用于线上服务监控、发布验证、回归测试等多个场景。

自动化压测。随着购物入口访问用户的高速增长，系统所面临的压力也急速增加，如何能高效、准确地评估系统的抗压能力成为团队需要重点考虑的事情。而性能测试本身是一件工程量极大，并且需要多方配合完成的工作，不仅资源占用多，沟通成本高，且对于复杂的场景，构造适合场景化的数据更是难上加难。

市面上的压测工具品类繁多，经过评估，团队最终选择了WebBench和TCPCopy两款工具用于压测执行，前者较为轻量，后者则能模拟线上真实用户流量，更贴近场景，两者相辅相成共同构建压测体系。压测系统将两款工具集成到自行搭建的Web页面中，提供简便的交互操作，同时提供自动化执行压测任务的能力。执行人员定制好压测计划后，系统即可按照设置好的任务、执行时间来定期执行压测任务，也可以在计划中将执行触发条件设置成当被测对象有变更的时候自动执行。由此，既减少了人工反复配置发起压测动作的工作量，也避免了每次配置中可能出现的人为错误，可以更方便准确地开展系统压测工作。

从压测执行原理来说，以TCPCopy为例，首先需要对被测对象定制压测任务，为了让压力更真实，团队选取的是采用线上流量来施压的方式，因此，工作的重点之一是真实流量的

录制。系统设计为在指定时间拉起流量录制脚本，并到目标机器搜集流量，然后保存到压测系统本地文件。流量录制结束，系统自动进行流量播放，并且智能地搜集后台相关性能数据。压测结束后，再基于收集到的性能数据做二次分析，输出压测报告，便于开发和测试人员分析可能存在的问题。由于采用的是线上流量进行压测，因此在数据的多样性、真实性上会比模拟制造出来的测试数据更符合线上的场景，从而也确保了压测的结果更加真实可信。

自动化安全扫描。微信手Q购物渠道由于依托强大的微信和手Q平台，业务形态具有强烈的SNS属性，更加注重好友间的互动、转发和分享。而Web端的安全攻击，有很大一部分是可以通过修改URL中的参数及参数值来完成的。加之面对海量的外网用户，电商场景中的用户资产、优惠券等也时常受到不法份子的觊觎，所以开发人员必须有能力和及时发现并封死这道暗黑之门。

上线前的安全测试是安全防范的重要手段，但是单纯依赖人工进行检查，难免会因为人员技能经验的不足，以及执行流程上的疏漏，发生检查不到位的情况，因此，需要有一套更完善更自动的机制来有效保证系统的安全性。基于此，测试团队提出了安全性测试自动化方案，通过分析线上用户访问所产生的Nginx日志，还原出原始URL请求，并针对这些URL请求创建并执行相关的自动化安全扫描任务。

一站式智能发布验证。很早之前在做自动化及各项工具建设的时候，团队就考虑过后续如何将建成的这些内容有机地集成到研发生命周期的各个环节中，尽量确保通过系统和流程要求触发执行，而非人为的随机执行，从而进一步确保产品上线质量，以及工具价值的最大化。例如，前端开发人员在发布前端页面的时候，发布系统就能触发前端页面性能扫描工具执行，并及时将报告输出给发布者，这样就能在第一时间知道页面性能状况，不需要等到测试阶段才发现问题。

沿着这样的思路，团队逐渐推进各项集成工作的实施，并先后完成了前端性能扫描工具及CGI自动化、后台接口自动化、安全测试工具与发布系统的集成，实现了一站式的智能发布测试验证。

小程序测试探索。众所周知，微信公众平台小程序是一种不需要下载安装即可使用的轻量应用，可以在微信内被便捷地获取和传播，这是一种新的开放能力；其传播特性、用户群体以及使用场景和微信手Q购物渠道所开展的业务有很高的重合度，因此微信手Q购物渠道也在第一时间跟进并推出了京东购物小程序。

经过四个大版本的迭代，目前的京东购物小程序已经可以满足绝大部分用户的购物场景和购物需求，更多的功能还在持续迭代完善中。那么，对于这样一个新鲜事物，测试团队又是如何快速地适应调整，确保它的上线质量的呢？由于也是在逐渐摸索尝试，以下只是简单介绍几种目前已在开展执行的手段。

◆ **静态代码的扫描。**由于小程序的源码语法接近于JavaScript，因此引入了JSHint进行源代码的静态扫描工作。静态代码扫描最关键的是扫描规则是否适用于自身团队，因此，将经验总结以及对应的开发规范加入到自定义扫描规则中进行扫描。例如，iOS8系统上对部分ECMAScript6特性（数组的Includes方法等）支持情况不同，导致for-in等语句运行时产生异常，也会对无效函数进行扫描。扫描结果以Git代码版本、代码文件、临近行MD5值等



维度进行记录，支持Web可视化的展示与修复反馈。静态代码扫描工作在转测试及上线等环节都会开展，第一时间暴露缺陷。

◆ **调试与性能测试平台。**小程序调试平台及小程序性能测试平台，通过在源代码中增加Hook函数，搜集上报运行过程中的业务和系统信息，同时支持用户资源的查看和修改，在很大程度上弥补了小程序开发工具以及体验账号等各方面的限制。并且这种方式可以将业务和系统信息落地存储，测试人员可以通过进一步的查看来发现一些和当前测试场景不直接相关的异常，最大限度地避免潜在问题暴露到线上。小程序性能测试平台则是另外一个优秀的测试平台，通过类似的技术方案，可以将小程序运行过程中客户端小程序进程所占用的CPU和内存信息收集上报，结合对应的操作日志，准确地定位到哪些操作消耗资源比较多，进而可以进行有效的、有针对性的分析和优化。

◆ **小程序自动化方面。**采用了Appium+Ride+Fiddler组合的技术方案，其中Appium作为底层的驱动框架，Ride用来维护和开发测试用例，通过Fiddler插件并内置自定义规则的方式，在测试计划运行过程中，出现HTTP状态码异常、返回包结构异常以及返回包大小超出限制等情况时，记录相应的会话数据，并根据用例所归属模块，针对性地通过邮件的方式推送给相关责任人进行确认修复并回归验证。随着基础用例的不断完善、升级和固化，目前小程序自动化用例已覆盖领券、秒杀、商品详情页、购物车和结算等主要场景，在保障转测质量、发布质量以及线上运营监控等方面都起到了不可替代的作用。

产品在不断迭代，业务在不断变化，作为质量保障团队，唯有紧跟前端团队的步伐，不断学习与优化，才能确保高质量高效率的支持好业务，同时保持自身团队的持续发展。



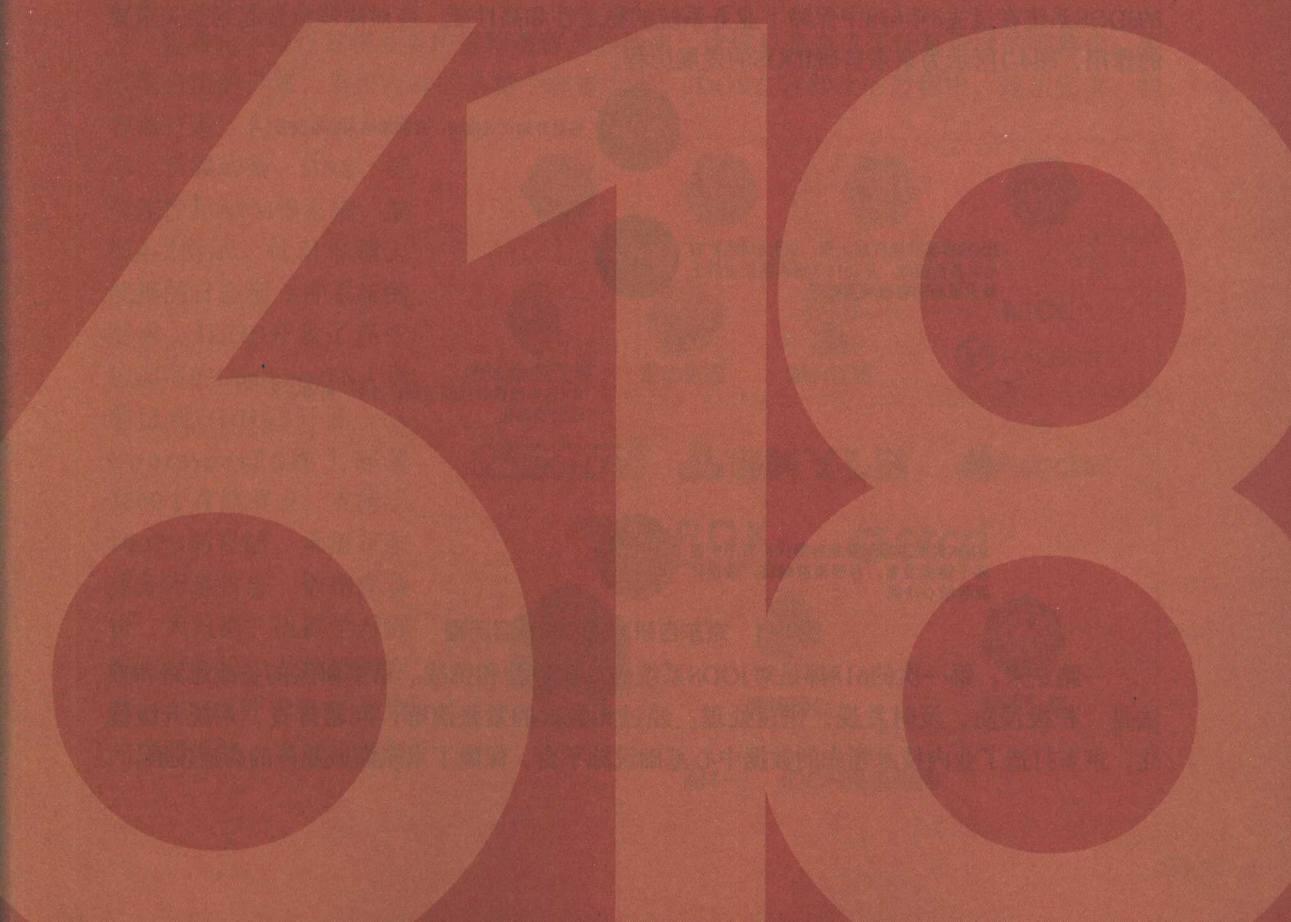
Cookie：微信手Q业务部研发中心总监，负责研发中心质量管理工作，具有丰富的电商测试及质量管理经验。

第4章

基础保障

坚实高效，安全稳固

万丈高楼平地起。只有坚实稳固、高效灵活的基础平台，才能支撑起京东千套业务系统，让百万核CPU计算资源，数千个存储节点能够为京东业务发展所用。





4.1 基础架构

2017年618已经落下帷幕。这个从京东店庆日演变而来的促销节，已经成为中国电商行业每年最为盛大的“节日”之一。因此，每年的618也成为技术人员的一场技术保卫战，从资源规划到夯实基础架构、增强智能，以及为确保系统稳定无故障的模拟演练和全链路压测。资源规划是大促备战的必要条件，可将资源规划系统分为平稳型和毛刺型。对于平稳型系统抽取70%的资源分而治之，高效分配；剩余的30%毛刺型系统的核心流程需充分扩容。京东为提升多数据中心架构的稳定性，实施了基于Docker的全面容器化，加以弹性计算能力提高资源利用率、扩容确保交付更快速等，自研中间件技术和以内存JIMDB为中心的存储体系，使用高性能微服务框架JSF接入实例，同时JMQ消息平台同步复制和刷盘确保数据不丢失来夯实基础。应用AI技术至商城各个业务环节，如订单取消及“冷静期”、订单预测等。破坏性地模拟线上故障，提升全员的应变能力，最后对于每个环节及地区实施黄金链路压测，筑成京东坚实的基础系统架构。

4.1.1 容器集群

京东618是一个狂欢的节日，GMV、系统流量屡创新高，在这背后是基础设施平台的技术人员在全力保障底层计算、网络、存储等资源和服务的稳定。

京东是国内最早开始在生产环境中大规模使用Docker容器技术的公司。特别是京东自研的JDOS系统在过去3年618中保障了业务系统的稳定性和高性能，应对高峰流量起到至关重要的作用，图4-1所示为京东自研JDOS的发展历程。



图4-1 京东自研JDOS的发展历程

一路走来，每一次的618都是对JDOS系统的一次检验和挑战。京东研发的备战思路和方法是：积极预防，及时发现，快速处理。经过无数次的紧张演练，问题排查，系统升级优化，京东打造了业内技术领先的数据中心基础设施平台，保障了京东商城业务的高速发展。

1.业务基石，自由计算

从2014年起，在其他公司还在观望调研容器技术的时候，京东已经在JDOS系统的帮助下率先进入了容器化时代。回想JDOS系统在刚开始研发的时候，从几个人的小团队开始起航，迅速掌握了OpenStack和Docker的核心代码。在业务推广的时候，每个团队成员对接一个核心业务，与业务团队一起调优。因为业内并没有成功的案例可以参考，所以JDOS团队只能通过自己的努力去解决这些问题。这个过程非常艰辛，团队成员每天不仅要完成代码开发工作，还要对Docker的每一个模块花大量的时间进行深入的研究。开始时，容器性能并不能满足部分0级系统业务的需求，为了解决这个问题，我们对Docker进行了反复地测试优化，不放过每一个可能出现问题的环节。经过几天夜以继日的持续改进优化，Docker容器终于通过了大规模长时间的压力和稳定性测试，TP99也降低到40ms以内的范围，满足了业务的要求，可以放心的上线了。在这一过程中，JDOS从系统内核到Docker，再到JVM进行了全方位的定制优化，以保证业务应用在容器中运行得到近乎物理机的性能。接下来业务开始逐步从运行在物理机、虚拟机中，迁移到容器中运行。容器化带来的不仅是计算效率的提高，更是计算资源供给能力的提升。在不增加计算资源的情况下，计算资源供给能力的提升需要依赖计算密度的提高。JDOS在保证安全和隔离的情况下，将不同业务的应用进行了彻底的混合部署，将计算资源进行了最大限度的复用。业务的扩容不必再局限于物理机的采购周期，而是通过业务分级，将计算资源进行调度和重新分配，以保证业务的计算需求得到满足。截止2017年618，京东商城JDOS平台管理着数百万核CPU的计算资源，是京东业务运行之基石。

容器化不仅仅是将业务的进程搬移到容器中，这样只是完成了计算资源的容器化。要实现自由的计算，需要打造的是整个容器生态。JDOS在容器化的过程中，逐步完善，为容器打造了包括存储、网络、负载均衡、DNS、智能监控和APM等系统，如图4-2所示。针对容器大规模的日志型文件系统的需求，JDOS开发了高效追加写的ContainerFS（目前已经GitHub开源）。ContainerFS线上部署3000个存储节点，支持了100PB的存储，承担京东业务海量日志、数据库备份、大数据、机器学习训练临时数据以及模型的存储。JDOS针对百万级别的容器网络，实现了高效

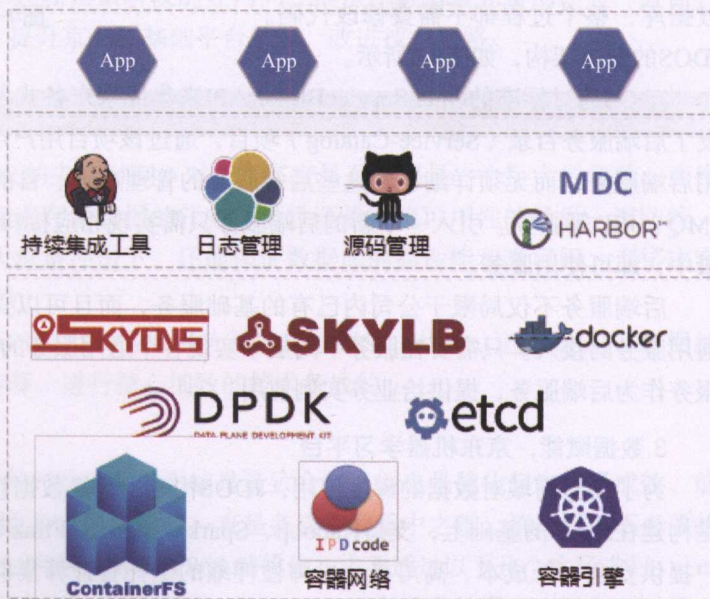


图4-2 JDOS系统架构



的Cane网络，压榨网卡的极限性能。特别针对网络中的小包转发，使用DPDK技术对负载均衡ContainerLB进行加速，使得万兆网卡达到14.35Mpps的硬件极限。监控也是JDOS的重要组成部分，为调度、排障、资源预警和规划提供强有力的数据支撑。JDOS研发了MDC监控系统，监控资源涵盖容器、物理机、交换机等，监控数据粒度达到2s，提供包含系统指标、网络指标、磁盘指标、硬件指标、容器指标五大类，共31个小类常用的指标监控服务。每天生产监控数据3200亿条，数据文件可达60TB。MDC还可以为用户提供在线的秒级实时监控服务，为故障排查提供及时可靠的数据支持。

2.一站式服务，京东PaaS平台

JDOS不仅是基础设施资源的管理平台，将京东基础设施资源进行统一的管控，而且也是应用服务融合的平台，为应用提供中间件等后端服务的资源管理服务。JDOS涵盖了从源码到镜像，再到上线部署的CI/CD全流程，提供从日志、监控、排障、终端、编排等一站式的功能。从这个意义上来说，JDOS也是京东的PaaS平台。

在JDOS的视角里，一切皆资源。JDOS把后端服务当作附加资源，与常见的负载均衡、容器等资源地位同等。JDOS认为后端服务是可以按需加载、卸载、替换的。例如，如果应用的数据库服务由于硬件问题出现异常，管理员可以从最近的备份中恢复一个数据库，卸载当前的问题数据库，然后加载新的数据库，整个过程都不需要修改代码。JDOS的整体架构，如图4-3所示。

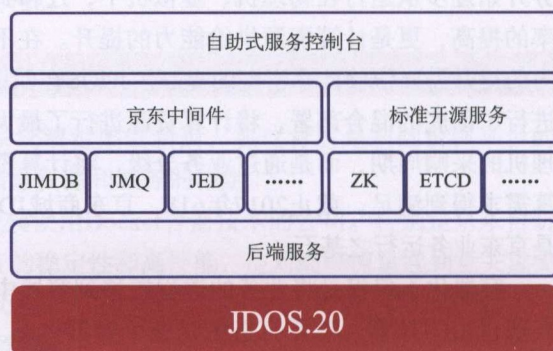


图4-3 JDOS架构图

JDOS通过标准的OpenServiceBrokerAPI来集成京东各式各样的后端服务。京东自行研发了后端服务目录（Service-Catalog）项目，通过该项目用户可以轻松地应用程序配置使用后端服务，而无须详细了解这些后端服务的管理逻辑。目前后端服务已经支持JIMDB、JMQ、JED等服务。引入一个新的后端服务只需实现相应接口，注册到JDOS的后端服务目录中，即可使用服务。

后端服务不仅局限于公司内已有的基础服务，而且可以支持其他如RabbitMQ/Redis等通用服务的接入。只需要在服务中间层中实现各个通用服务的标准API，即可接入其他通用服务作为后端服务，提供给业务实例使用。

3.数据赋能，京东机器学习平台

为了助力商城对数据的深度使用，JDOS构建了弹性数据计算平台。弹性数据计算平台是构建在JDOS的基础上，支持Hadoop、Spark、Storm、Flink等计算框架的生态系统，为用户提供安全、低成本、高可靠、可弹性伸缩的全托管计算集群，提供集群、作业、数据等管理的一站式大数据处理分析服务。

弹性计算平台不仅支持现有主流的大数据计算框架，而且还支持机器学习框架。现在市场上的一些平台，提供了算法拖曳等功能。这些功能对于机器学习的初学者来说，有助于其对一些基本算法的加深理解和认识。因此这些平台主要为用户提供小型低配的容器或者虚拟机，供其学习使用，然而这些平台对于计算的效率和平台的稳定需求并不是很高。

不同于这些面向入门级用户的平台，JDOS的机器学习平台主要面向的是具有编程技能的高级用户，特别是对计算资源（CPU/GPU）有大量需求、进行工业级生产的用户。JDOS的优势在于能够在秒级提供从数百核到数十万核的计算能力。充足的计算能力和稳定高效的基础设施管理为机器学习提供了稳健的平台保障。

用户在使用深度学习服务时，需要提供深度学习的训练脚本，并上传到平台的存储中。用户配置参与分布式训练的参数服务器数量、工作服务器数量等参数，将参数提交给平台。平台将对参数进行统一的分解，根据内置的TensorFlow、Caffe模板生成对应的容器、负载均衡等资源，并最终提交到JDOS平台进行相应的创建。用户无须对参数服务器、工作服务器进行控制。平台会统一管控，当容器失效时会自动进行迁移重建。最终的训练结果也将保存到平台的存储中。用户可以通过对训练的实时监控，查看相应的训练日志等掌握训练的状态。

容器平台实践之路，京东不仅走得比较早，也走得比较坚定。每次618的业务规模增长都在驱动着系统的不断升级。不畏将来，不念过往，京东对技术永怀敬畏之心。未来，JDOS系统将会继续演进，并且会将京东基础平台系统的底层技术开源，把在实践过程中的很多心得和技术视野分享出来。比如认为容器本质是Linux Kernel技术，容器需要数据中心底层基础软件全力配合，如分布式域名解析、高性能负载均衡、分布式共享存储、精确授时等。因此京东在这些方面不希望闭门造车，而是能够和更多的业界人士来分享经验与教训。一方面，为更多底层技术还在摸索阶段的业内同仁提供一点借鉴和帮助；另一方面，也希望获取业界专家的指导，提升京东的基础平台系统，改进技术思路。

4.1.2 数据库

每年京东618，特别是零点的尖峰时刻，对于京东交易系统都是一个巨大的考验。而作为最后端的数据库，不仅承担着存储数据的任务，而且还是系统可用性的最后一道防线，如何在短时间、高并发的超大流量冲击下，还能保证数据库的高性能和高可用，直接决定了整个618期间系统保障的成败。

本节将详细介绍在每次618前，数据库团队是如何通过SQL优化、迁移改造、数据结转、架构优化、应急值守等步骤，进行精心细致的技术备战的。

1. SQL优化

众所周知，低效的SQL语句对数据库的影响是最致命的，但也是优化起来最具成效、能最快解决系统性能问题的，因此SQL语句优化一直是备战中的重中之重，消灭所有不必要的慢SQL是DBA永远的目标。DBA和研发对于慢SQL的优化，主要通过以下几个途径进行：

◆ **慢SQL平台**。京东数据库团队使用基于Anemometer二次开发的慢SQL平台，对线上



慢SQL进行自动化、可视化的管理。

◆ **邮件推送。**慢SQL平台虽然可以比较直观地显示系统的SQL语句，但访问是被动的，如果研发或者DBA没有主动登录慢SQL平台进行查看，就无法及时发现慢SQL。因此针对某些重点系统，团队编写了脚本进行主动邮件推送，如果发现有慢SQL产生，每天会定时发邮件给相关研发，提醒他们及时进行关注和优化，并且会把慢SQL提交Jira系统，进行有效的跟踪，直至该慢SQL被优化完成。

当前京东数据库团队也在开发新一代的数据库优化平台，除了可以更智能、自动、直观地把慢SQL推送给DBA和相关研发外，还能够基于一定的规则进行自动优化，远期目标是达到智能优化。

2. 扩容升级

每年都会根据上一年度618的峰值数据，再结合当年业务部门的销售计划，进行数据库的容量规划，通常来说都是按照上一年度该数据库QPS的5~10倍进行准备。升级改造对于DBA而言，是最头痛的事情，因为要涉及大量的系统进行硬件扩容，数据迁移等烦琐的工作，而且操作上具有很高的风险性。

3. 数据结转

所谓数据结转，其实就是历史数据归档，宏观来说就是数据生命周期管理，对于数据库而言尤其重要，因为数据库是对I/O异常敏感的系统，减少存储的数据量和每次I/O读取量，对数据库性能都有极大的益处。因此每次618前，数据库团队都会对线上数据库进行历史数据归档。通常的做法是，把历史数据先转移到历史库，确认历史数据拷贝无误后，然后在主库进行删除操作。

4. 架构优化

经过SQL优化，数据结转，扩容改造等优化措施之后，如果通过压测演练发现还是有问题，这个时候可能就需要从整体架构和业务流程上去考虑优化，目前的架构是否合理。从数据库架构方面考虑，一般推荐下面的做法：

- ◆ 对于高并发频繁读写的数据，前端部署Redis等内存数据库进行Cache。
- ◆ 日志类和大字段数据迁移出MySQL，采用MongoDB、Cassandra等NoSQL或者JFS进行存储。
- ◆ 复杂的查询，将数据迁移到Solr、ElasticSearch、Hadoop等平台。

5. 应急值守

以上所有的优化手段都做完后，基本也就进入618期间了，这个时段DBA基本处于待命的状态，值班期间的具体要求如下：

- ◆ **24h值班。**虽然大量的操作都已经自动化，但是很多故障还是需要DBA在现场，第一时间进行应急处理。
- ◆ **严格控制上线。**很多故障都是由于上线导致的，因此618期间会严格限制上线，除非修改必要的Bug或者特殊情况。
- ◆ **多维度审核操作。**必要的上线，也要经过多维度审核才可以，首先研发团队会进行

评估，然后数据库团队也会进行评估，最后还需要总监以上级别领导批准，然后才可以双人上线操作。

◆ **集中办公。**618期间，主要研发和DBA都会在一个会议室办公，出现问题可以及时沟通。

◆ **自动化运维平台。**京东数据库团队自主研发的JDBS自动化运维平台，可以进行数据库自动化运维、监控智能分析、部分故障自动处理等，来辅助DBA进行备战和值守。

每年的618，对于DBA来说都是一次极大的考验，也正是这一次次的考验，使得京东的数据库不论走运维自动化，还是走数据库架构、性能优化，都日趋完美，也更加安全、稳定、高效。

4.1.3 中间件

中间件系统作为京东集团技术体系的基础服务，被广泛应用在商城、金融、物流、京东云等子公司中，中间件贯穿京东大部分的业务场景，包括采销、订单、支付、库房、分拣、配送、财务等。中间件产品主要包括分布式服务框架JSF、分布式消息中间件JMQ、分布式内存存储JIMDB。

随着业务的不断发展，中间件系统不断进行改进和完善。现在通过JSF发布的接口有几个万个，每天JMQ处理的消息近千亿条，JIMDB维护了近10万个Docker实例。下面重点介绍一下这几个产品的发展历程以及一些技术方案。

1.JSF

◆ **第一版JSF。**2012年初，京东从.NET转Java。各个部门，各个业务线都没有一个统一的服务化框架，有的使用Dubbo，有的使用WebService，有的使用Hessian等。

由于业务规模的原因，系统服务化的观念还比较薄弱，系统间的接口比较少，拓扑相对比较简单，通过统计接口规模在1K左右，服务节点在50K左右。服务部署的机房之间相隔很近，又有专线连接，因此当时也不需要考虑跨机房的问题，基于当时的情况，第一版服务框架的愿景和目标比较明确。

①京东系统服务化、API化的从无到有。

②统一京东的RPC调用框架。

③稳定可靠。

④提供简单的服务治理功能。

根据当时的实际情况和业务需求，研发团队选取的技术方案选择如下。

①RPC框架：基于Dubbo2.3.2做配置扩展和功能扩展，包括Rest（Resteasy）、WebService（CXF）、Kryo/Thrift序列化、调用压缩等。

②注册中心：采用ZooKeeper。

③监控中心：监控服务+HBase。

④管理平台：读取ZooKeeper做管理平台，提供基本的上下线、黑白名单等功能于2012年4月上线，最大规模时，接口数3K，接入最大IP数20K。



◆ **第二版JSF**。随着京东业务的快速增长，接口、业务实例数呈数量级增长。同时京东成立子公司，在全国各地新建机房，部署结构也变得更加复杂。加上SAF遗留的一些问题，大概面临如下挑战：

- ①RPC框架较重，性能有提高的空间。
- ②注册中心无业务逻辑，直接对外暴露，不方便扩展和降级。
- ③京东复杂的部署架构需要更强大灵活的服务治理功能。
- ④监控信息可以进一步完善。
- ⑤没有统计应用之间的依赖关系。
- ⑥有跨语言调用需求，但是现有框架不支持。

随业务规模不断扩大，在2014年初，团队开始自研新一代的服务框架。主要做了如下技术选型：

- ①RPC框架：轻量级，更佳的性能，兼容旧版本协议。
 - ②注册中心：基于DB作为数据源进行持久化，前置Index服务进行注册中心路由；把一些逻辑放在注册中心减少客户端负担。
 - ③监控中心：监控数据存放在InfluxDB中，后来又改为ElasticSearch。
 - ④管理端：订阅关系，黑白名单，限流等元数据存放在DB中，相比ZooKeeper更加灵活方便，可以提供完善的服务治理管理功能；通过与公司内部的应用管理平台联通，可以描绘清晰的应用依赖关系。
 - ⑤HTTP网关：基于Netty开发，提供HTTP协议到内置协议的转换，支持跨语言调用。
- 系统架构如图4-4所示。

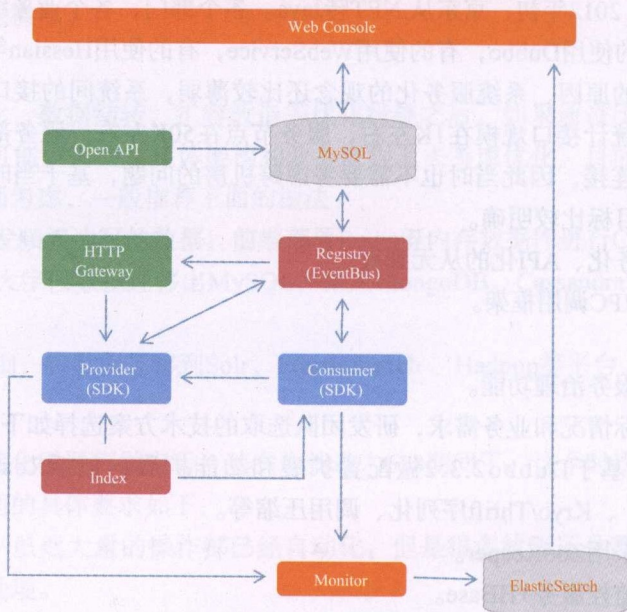


图4-4 JSF架构图

新一代JSF上线后，注册中心可以很方便地扩容，每个机房都可以独立部署到一个注册

中，系统的服务能力得到了很大提高，JSF服务的接口涉及的实例达到几百万个。

随后由于京东加快了Docker化部署的步伐，大部分应用部署到了带有独立IP的容器中，IP数瞬间就增加了一个数量级，旧版本的JSF平台使用ZooKeeper作为注册中心，业务压力非常大。618时，由于会有一些服务集中进行重启，导致ZooKeeper上的写入量突增，读写请求处理发生阻塞，应用服务获取不到接口服务的信息，无法进行访问。开始由于事情比较紧急，业务应用的开发人员根据以往的经验不停地重启服务。越是访问不通，越是批量重启，最终导致雪崩，当定位到问题后，由于ZooKeeper直接暴露服务，无法降级，拿不到接口信息的服务会重复查询，ZooKeeper重启后很快宕机，无法提供服务，最后只能通知业务方改用直连的方式缓解这个问题。经历这次事件后，各业务团队认识到版本升级的重要性，但是代价沉重。由于旧版本经历过注册中心事故，下面重点介绍新版的注册中心从架构上是怎样应对大规模访问的，对横向扩容做了哪些改进。

在服务框架中，注册中心是一个非常重要的模块，提供服务列表的注册、订阅、推送，服务配置的变更通知，服务状态的实时查看等功能。

注册中心实例自身是无状态、可水平扩展的，服务列表和配置等信息会保存在DB中，注册中心的内存中会保留一份完整的最新的信息。

每个机房部署多个注册中心节点。同机房的RPC框架会优先连接本机房的注册中心节点。由于注册中心需要频繁地与服务的提供端和服务的调用端进行通信，因此实例调整和扩容会相对频繁一些，为了保证注册中心地址变更对业务系统透明，有一个专门用于注册中心发现的服务（Index），它是一个HTTP服务端，只用于提供最新的注册中心节点拓扑信息，服务简单，访问压力小，不会轻易变动。

目前JSF已经平稳运行了2年多的时间，虽然有了一些小的Bug，但是没再发生系统性的事故。每天通过JSF接口服务访问几千亿次，为几万个接口提供服务。

2.JMQ

2012年初，京东有一个基于数据库开发的消息分发系统，消息发生堆积时会导致性能变差，而且不方便扩容。因此建设一个统一的消息平台被提上了日程，目标就是提供高可用，可扩展的消息传递服务，并且能够对消息接入系统进行有效的治理。

第一版。消息服务：基于ActiveMQ5.6做扩展。集群配置中心：MySQL+ZooKeeper。管理监控平台：自主研发，提供比ActiveMQ自带的管理平台更丰富的消息监控功能，并提供消息管理服务。

研发团队对客户端进行了封装，几组Broker构成一个Topic集群，一组Broker上可以分配多个Topic。客户端发送时可以随机发往集群中的一组Broker，消费端会连接所有的Broker组进行消费。当服务端发生故障时，客户端会重试发往集群下的其他的Broker组。

通过一系列的改造，基本实现了消息平台的高可用性和高可靠性，但是也发现了一些性能问题，主要在以下两个方面进行了改造来提升性能。

原生的ActiveMQ是通过消息转发的方式，在Slave上需要同步对索引进行操作，消息需要多次序列化、反序列化，为了提高复制性能，对复制模块进行了重写，基于Journal进行复制，在Slave端异步构建索引。



ActiveMQ通过虚拟主题（VirtualTopic）的方式实现Topic，即如果一个Topic有多个订阅者，ActiveMQ就为每个订阅者创建一个队列，那么Producer每发一条消息，有多少个Consumer，Broker就会将消息复制多少份，将其放到不同订阅者的队列中，Broker再从中获得消息推送给Consumer。在这个问题上我们遇到了一些性能问题，有些Topic有几十上百个Consumer，由于是同步刷盘，因此刷盘次数被放大几十上百倍。为了提高性能进行了改造，即把多条相同消息写入后只刷盘一次，针对这类场景性能提升4倍以上。

ActiveMQ的架构方案，在性能上有进一步优化的空间，比如每次消费消息进行ACK的时候，服务端都需要操作索引，发送消息也需要操作索引，相互之间会竞争，VirtualTopic需要拷贝消息等，另外当需要添加新的特性时，比如重置消费位置进行消息回放，在现有代码的基础上进行修改变得不太可能，修改困难很大，基于以上原因，在2014年初决定全部重写，开发一个全新的消息平台。

第二版。在上一版的基础上主要做了以下一些工作。

◆ **服务端：**实现轻量级的存储模型、支持消息回放、消费异常支持重试、支持消息轨迹、支持顺序消息、支持广播消息等，并兼容ActiveMQ客户端使用的OpenWire协议。

◆ **客户端：**实现轻量级客户端只和Broker通信，支持动态接收参数，内置性能采集、支持就近访问等特点。

◆ **管理控制平台：**丰富了管理监控功能。

◆ **HTTP代理：**基于Netty开发了对HTTP协议的支持，支持跨语言访问的能力。

在一些关键的设计方案上有以下几个特点：

◆ 新版本没有采用ActiveMQ的方式实现网络通信，而是使用了Netty 4.0，此框架开源，支持epoll，编程模型相对简单。这在一定程度上减少了服务端的开发工作，也降低了服务端的复杂度。

◆ 相比ActiveMQ存储层做了重大的架构调整，存储层包含三类文件，分别为日志文件（journal）、消息队列文件（queue）和消费位置文件（offset）。

日志文件，主要存储消息内容，包括消息所在队列文件的位置，有以下特点：

- ①同一Broker上不同Topic消息存储在同一日志文件上，日志文件按固定大小切分。
- ②文件名为起始全局偏移量。
- ③消息顺序追加。
- ④日志文件同步刷盘。

由于JMQ主要使用在可靠性要求极高的下单、支付等环节，所以Broker必须保证收到的每条消息都写入物理磁盘，这样的日志文件设计主要是为了提高多Topic大并发下磁盘的写性能。为了提高写性能，研发团队在逻辑上还实现了Group Commit（组提交）。

JMQ消息需要实时写入磁盘中，目前JMQ大部分服务运行在SAS盘上，带宽有限，消息队列之间很容易相互干扰。开始消息队列没有按照业务部门，重要程度等进行隔离，618的时候，有的消息流量突增，分配在相同物理机上的其他队列发送延迟时间TP99飙升。为了避免这类情况出现，根据消息队列的情况做了物理上的隔离，避免互相影响，针对性能要求特别高的消息队列，比如订单消息、支付消息就独立使用资源。

JMQ在运维当中，碰到的更多的问题是，消息的消费者由于应用问题，或者这个消息对应用不重要，处理的资源比较少而发生消息堆积，消息存放的文件由于不能被清理导致物理机存储报警。JMQ虽然在公司被业务应用广泛使用，取得了很大的成功，但是在大数据处理环节，大部分的应用在使用Kafka，接下来JMQ有计划支持Kafka协议，使其在大数据生态环节中发挥更大作用。

3.JIMDB

开始，直接访问数据库就能支撑系统的流量，系统架构也很简单。当流量一天天增大，慢慢地有些业务开始使用Memcached，后来由于Redis支持丰富的数据结构，使用越来越广泛，逐渐在京东内替换掉了Memcached。随着Redis规模的扩大，碰到了很多运维上的痛点，因此决定基于Redis进行二次开发，实现集群功能，解决扩容和故障恢复等问题，也就是现在的JIMDB平台。

第一版。2014年初决定做一个缓存的平台，主要解决以下几个问题：精确的故障检测和自动故障切换，无损扩容，以及提供监控和报警等服务。

◆ **故障检测。**在故障检测和故障切换的方案中，比较容易想到的方案就是引入ZooKeeper，通过ZooKeeper的临时节点探测不存活的服务，但是由于需要修改服务端代码、不方便跨机房部署、watch数目和连接数过多等性能问题，最终没有被采用。京东决定研发一个探测程序，这个探测程序主要是检测JIMDB Server实例的存活状态，但是它需要尽可能地解决由于部分网络不通时导致的误判。采用的方案就是，部署多个探测程序，每个部署在机房的不同机架下。多个探测实例同时对一个JIMDB Server实例进行探测，当有一个探测实例检测到JIMDB Server实例是存活的，那么该实例就认为是存活状态，当没有反馈其为存活状态，且超过半数的探测实例认为该实例死亡时，则通知故障恢复程序进行主从切换，变更集群拓扑结构，并把新的拓扑结构通知给所有的客户端。故障检测和恢复的问题算是解决了，接下来就是解决扩容的问题。

◆ **无损扩容。**业务在刚上线阶段，访问量和需要缓存的数据量并不大，提前申请很多的缓存服务会导致资源的浪费，随着业务的增长或者需要搞活动促销时，访问量和数据变大，缓存服务端需要扩容。

为了解决以上问题，开始对服务端进行改造，服务端引入slot的概念，数据按照slot进行组织，一个服务端实例可以服务多个slot，客户端中维持一份与服务端一致的slot信息。由于服务端有了slot信息，因此可以判断那个Key是否在自己的服务范围内，还是已经迁移出去了，可以避免数据被写错。有了slot，迁移时服务端以slot为单位进行数据迁移，避免了从全部数据中找出应该迁移的数据，另外直接通过待扩容的服务端往新扩容的服务端写数据，避免数据经过过滤层的转发，大大提高了迁移速度。

第二版。随着集群规模进一步扩大，使用的业务越来越多，很多以前注意不到的问题逐渐暴露出来。

◆ **自动弹性调度。**业务流量突然飙升，容量不足报警等都需要运维通过管理工具进行扩容增加实例数，另外也有一部分业务申请了集群空间，由于业务调整等原因，访问量变小或者停用，平台管理人员比较难发现。为了提高平台自动化能力，减少运维人员的工作



量，需要让平台动起来，弹性伸缩的需求摆在了开发人员的面前。

为了让平台实现弹性伸缩，需要对集群的各项指标进行监控，比如OPS、内存使用率、网络流量等进行监控，统计这些指标一段时间内是否达到了设置的阈值，当超过扩容的阈值时则自动触发扩容，当低于缩容的阈值时自动进行缩容，释放资源。

缩容的过程和扩容的过程基本一致，把一个shard实例上的所有slot迁移到另一个实例上，当slot迁移完成后缩容也就完成了。扩容时就是把一个实例上的一部分slot迁移到一个新的实例上去。

扩容时由于需要增加实例，增加的实例应该部署在哪台机器上才合适呢，为了择出最优的机器，有一个采集程序会定期进行信息收集，然后根据CPU繁忙情况、网络流量、OPS、内存剩余空间、机器上的实例数等进行综合打分，各项指标都比较空闲的得高分，如果有一项指标不符合部署要求则直接淘汰，然后再从得分高的机器中选择一台机器进行部署。由于扩容在集群中是并发进行的，因此有可能会被多个处理线程同时拿到同一台机器进行实例部署，当部署完成后可能实例数等指标就不符合要求了，因此需要有一个预分配资源的计算，对未使用的资源进行预占并被计算在内，如果部署失败，则需要把这些资源值做相应的扣除，避免并发部署资源超限的情况出现。

对同一个集群还需要控制每台物理机上最大可部署的实例数，避免同一个物理机部署实例数过多，当机器故障时对同一个集群影响过大。

为了防止同一个路由器故障或者断电等情况的出现，同一个shard的主从实例应该跨机架，对有跨机房需求的应用，主从还应该部署在不同的机房。

◆ **服务端升级。**当一个平台需要维护成千上万个进程，这些进程在升级过程中不能中断服务，而且这些进程还是有状态的，必须保证数据正确无丢失，不能通过简单的流量切换就解决问题，升级程序成为了一项富有挑战的事情，同一台机器由于服务端升级需要维护多个版本同时运行，程序运行文件的分发和维护也变得不可控，容易引起混淆，如果由运维人员手动处理，不仅时间长，而且容易出错。

为了解决版本控制和快速创建和销毁服务引入了Docker，通过Docker的registry复制服务端版本管理和分发。通过启动一个Docker容器创建一个JIMDB的服务端实例。

程序升级过程中涉及数据的拷贝，为了加快升级速度，考虑到升级完成后旧的实例就可以销毁，因此在同一台物理机上创建一个新版本的实例，当旧实例销毁后并不会导致物理机上实例数超限等问题，通过数据复制流程把旧实例上的slot全部迁移到新的实例上，由于数据是在同一个物理机上流动，因此速度会比网络传输快很多。多个集群可以同时进行，也不会导致网络流量由于数据的迁移而暴涨。现在的JIMDB架构如图4-5所示。

JIMDB平台在一开始运行的时候，初衷就是让扩容对业务透明，业务不需要关心容量问题，很多业务系统并不关心存储的数据是否需要及时清理，避免空间浪费。另外有一些业务由于时间紧迫或者简单的架构，就把全量的业务数据都存放在JIMDB中，并没有进行冷热数据的分离。针对这些情况，接下来一方面就是加强沟通，督促业务清理掉不需要的数据，另外就是提供新的产品，对冷热数据进行区分，在维持接口不变的情况下，让热数据存放在内存中，冷数据存放在磁盘中，节约成本。

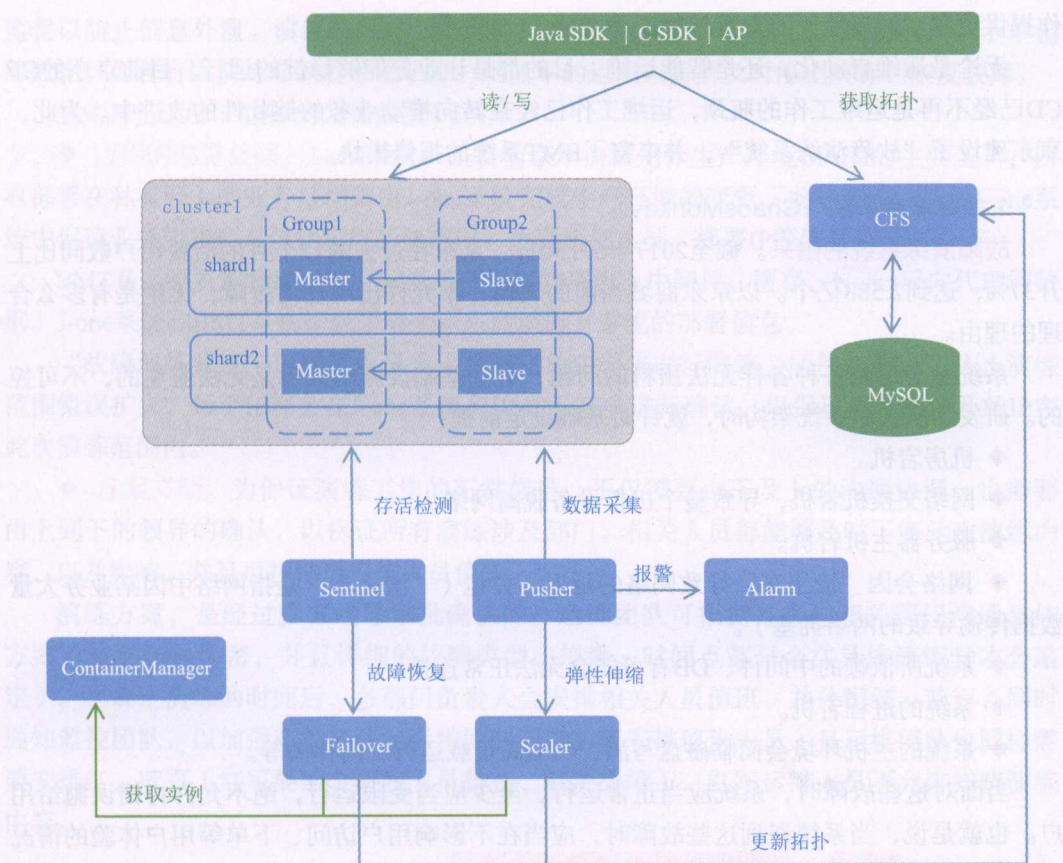


图4-5 JIMDB架构图

中间件产品在一定程度上已经取得了很好的成绩，成功支撑了公司各个业务系统的发展。不过系统还是有很多的发展空间，可以从以下几个方面进行：

- ①支撑的业务场景比较单一，没有与其他系统有机地结合，比如JMQ可以提供StormAPI，或者兼容Kafka协议，接入大数据处理平台。
- ②支持多语言，比如JSF只支持Java，对C/C++、Python等没有提供支持。
- ③规范使用，优化性能等，给业务应用提供架构支持，合理使用中间件资源。

4.1.4 智能运维

当前“机器学习”“人工智能”等概念方兴未艾，新技术的快速发展，既给运维工作带来了挑战，也带来了巨大的机会。

现在运维工作，已经不再是装系统、装软件、上线代码等传统型工作，智能运维应当通过多种方式、手段来降低企业在维护工作中的投入、减少人工元素的参与，并且有多种工具来自动完成故障的发现、处理与解决。目前京东商城在运维体系建设中，将运维自动化放在首要的位置，将人从重复、繁杂的工作中解放出来。运维人员的工作将从“操作者”转变为“决断者”。智能运维将是在“自动化运维”之上，为运维人员的“决断”工



作提供更多“纯理性”的方案与建议。

无论是运维自动化，还是智能运维，目的都是让业务能够稳健的运行。目前京东的CI/CD已经不再是运维工作的瓶颈，运维工作已经在转向推动业务的健壮性的改进中。为此，京东建设了“故障演练系统”，并丰富了BMT系统的报修模块。

1.故障演练系统（ChaosMonkey）

故障演练系统的由来。截至2017年6月30日，京东在过去的12个月的活跃用户数同比上升37%，达到2.583亿个。以京东商城当前的规模，不允许出现任何故障，无论是有多么合理的理由。

系统总会遇到各种各样无法预料的问题，有些基础设施的故障是无法避免的，不可控的。研发团队在做系统架构时，就针对基础设定前提：

- ◆ 机房宕机。
- ◆ 网络交换机宕机，导致整个机柜设备脱离网络。
- ◆ 服务器主机宕机。
- ◆ 网络会因“渣土车”导致网络的延迟、丢包（“渣土车”是指网络中因某业务大量数据传输导致的网络拥塞）。
- ◆ 系统所依赖的中间件、DB有可能会无法正常连接。
- ◆ 系统的进程宕机。
- ◆ 系统的主机环境会面临磁盘写满、CPU高负载运行、内存满等。

当面对这些故障时，系统应当正常运行，至少应当受限运行，绝不允许将错误抛给用户。也就是说，当系统遇到这些故障时，应当在不影响用户访问、下单等用户体验的情况下正常运行。

虽然系统在设计之初会考虑这些故障，但是故障真实发生时系统是否会达到预期，这是需要检验的。

2016年，研发团队开始做“故障演练系统”，也就是“ChaosMonkey”系统。通过在线上真实模拟所有可能发生的故障，来检验系统的健壮性。京东从2015年开始，绝大部分业务系统都已经迁移到京东内部私有云（JDOS）中，只有部分中间件系统、DB等还部署在物理机上。基础环境的统一，极大地方便了底层系统的开发工作。

故障演练的场景。业务系统做故障演练，要求是在业务量最低时。因为是线上真实演练，如若业务系统无法及时解决，有可能会对用户产生影响。为将产生的影响降到最低，一般都会在各系统的访问量最少时开始演练。

同时还要考虑到保密，即不能让业务系统的开发、维护人员知道在哪个时间点、哪个IP、会产生何种故障。在真实演练时，演练人员会分成两个“军种”：红军、蓝军。蓝军由各系统的主要负责人、运维人员构成，集中在一个封闭会议内，负责与运维人员共同“生产故障”。红军则由各系统的其他负责人组成，不限制具体办公地点，负责解决故障。红军与蓝军只允许通过对讲机进行沟通，目的是当红军遇到无法解决的故障时，蓝军要及时消除故障。

通常，会有一个较高级别的研发负责人“驻扎”在蓝军办公室内，做全局指挥，并且

监督以防止信息外泄。演练的同时，负责监控的同事，会时刻关注用户的访问量、下单量等数据。如若有因演练导致的单量下降，则蓝军的运维人需要及时停止演练、恢复故障。

整体架构设计。整体的架构如图4-6所示。

◆ **应用的部署数据。**J-one系统，是目前京东商城用作生产环境CI/CD的主要工具，所有部署在私有云上的业务应用都由J-one系统完成生产环境的变更、发布工作。所以J-one系统中保留业务应用所有包括应用所属部门、负责研发人员、部署IP等信息。

BMT是运维资源管理系统，记录了部署在物理机上中间件、缓存、Nginx反向代理等数据。J-one系统和BMT系统记录了整个京东商城所有系统的部署信息。

“故障演练系统”可通过由研发人员提供的IP列表进行演练。即使如此，为防止演练范围错误扩大，每个IP都会在J-one系统和BMP系统中进行确认，以保证演练所涉及的IP在此次演练范围内。

◆ **方案文档。**为保证演练工作的正常推进，不仅需要由下及上的沟通协调，也需要由上到下的领导的确认，以保证所有演练涉及部门、相关人员都能够及时了解此次演练内容，以及影响，并且可视情况安排人员值班。

演练方案，是经过多方领导审批确认的，运维团队可根据方案与相关部门沟通具体方案（需要加强保密，并且详细的故障类型、规模、时间点都只会在具体演练时才会敲定）。在确定演练的时间后，各部门负责人会安排相关人员值班，并分配红、蓝军，同时通知监控团队，以加强监控准备。运维团队同样需要安排值班人员，且运维团队也同样需要安排红、蓝军（红军要配合研发人员排查、解决故障），红军运维人员不允许知晓演练内容。

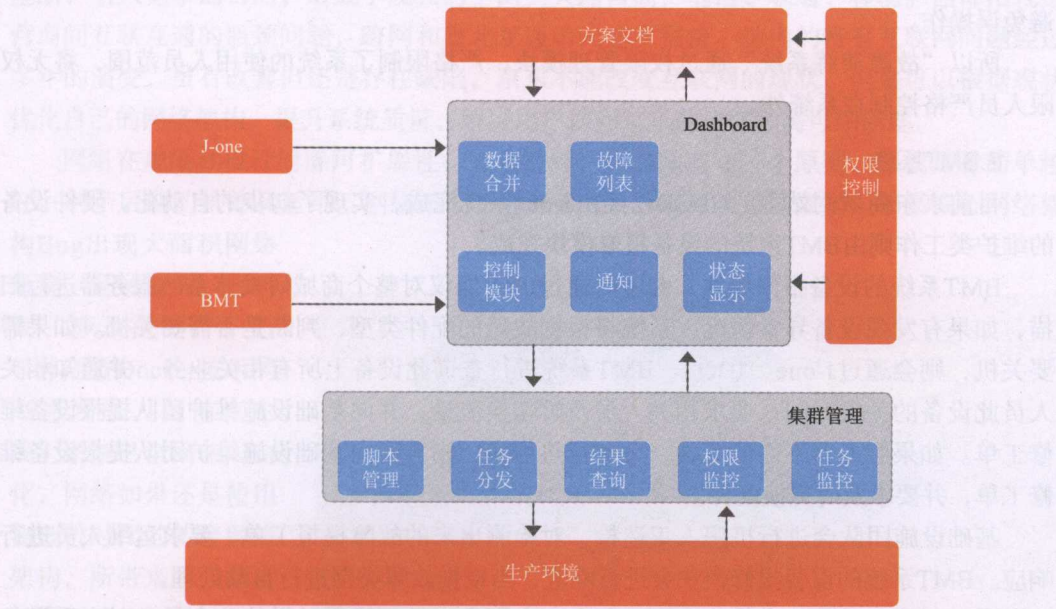


图4-6 故障演练架构图

◆ **Dashboard。**Dashboard页面，通过业务应用、IP两个维度进入故障演练操作。运维



人员通过选择应用，或者直接输入IP列表，开始演练操作。此时，运维人员可选择演练的故障类型、演练的规模、时间，设置完成后即可提交任务。此时，运维人员可在页面中看到当前任务涉及的IP，以及在此IP上具体的故障类型。

运维结果，当收到有故障影响到业务生产时，可在相应的任务设备中，执行“故障修复”功能，即可及时修复故障，恢复业务生产。

整个Dashboard，出于保密等需求，目前只对运维人员开放。

◆ **集群管理**，“故障演练系统”的核心功能。因为京东商城目前已经完全进入企业私有云时代。目前由商城研发系统—基础平台部负责的京东JDOS系统，承担着整个京东商城网站的基础环境支持功能。

针对研发人员来讲，已经不再关心具体的物理机、交换机等设备问题，研发人员所需要关心的只是其应用所使用的IP信息，以及此Docker的规格。因此故障演练系统大部分的演练单位都是以IP为标记的Docker。

很多故障，都是通过将生产故障的脚本推到Docker内，执行脚本来模拟故障。比如模拟“磁盘空间写满”故障，即是通过将脚本推到Docker中，执行类似“`ddif=/dev/zeroof=/path/filename`”命令完成。在恢复故障时，只需删除此文件即可。

通过集群管理模块，可以将任何可模拟故障的脚本推入到容器中并执行，进而在相应的容器上模拟相应的故障。但是有些故障的适用场景却受基础环境的限制，比如模拟内存空间使用满的故障，可能会出现容器因内存溢出而僵死的情况，进而无法恢复。所以内存空间使用满的故障目前只适用于物理机。

◆ **权限管理**。当前权限管理的主要内容还是确保系统的保密性：不让演练细节外露、避免误操作。

所以“故障演练系统”通过权限管理模块，严格限制了系统的使用人员范围，将无权限人员严格控制在系统外。

2.BMT系统

目前京东商城网站等的CI/CD已经由J-one系统完成，实现了初步的自动化。硬件设备的维护类工作则由BMT系统的设备报修模块完成。

BMT系统的设备报修模块，会定时通过IPMI协议对整个商城研发体系的服务器进行扫描，如果有发现设备异常情况，系统将根据故障的配件类型，判断是否需要关机。如果需要关机，则会通过J-one、JDOS、BMT系统进行查询此设备上所有相关业务，并通知相关人员此设备的关机时间，要求相关人员及时切换流量，并向基础设施维护团队提报设备维修工单。如果服务器不需要关机、可进行热更换，则自动向基础设施维护团队提报设备维修工单，并要求及时更换配件。

基础设施团队会进行机房人工巡检，对检测出来的故障提报工单，要求运维人员进行响应。BMT系统的设备报修模块会进行响应，当根据故障类型进行自动处理。

同时开放了运维的微信公众号，所有京东研发人员都可通过微信公众号及时查看设备的维修状态。

这套自动报修系统并未达到理想状态。目前仍然有一些系统无法做到对宕机故障的无

操作冗余，即系统遇到宕机故障时仍需要人为切换流量操作。持续自动报修系统会跟相关系统对接，实现无人值守报修。

目前团队只是做到了运维工作的自动化，运维智能化的工作也已经展开：通过收集的大量日志，将日志进行格式化，并进入机器学习平台进行训练，能够对研发人员、运维人员在故障排查时提供指导性建议，减少在故障排查时花费的时间、提高故障定位的准确度。通过对流量等数据的分析，为资源的使用提供合理建议，提高资源利用率。智能化运维，京东还在继续深入探索挖掘中。

4.1.5 异地多活

为什么要做异地多活？相信大部分企业开始考虑做异地多活时，绝大多数是因为企业的发展所驱动，而不是跟风潮流，按照企业的生命周期理论，一般会经历四个阶段：创业期、成长期、成熟期和持续发展期。所有的互联网公司在最初创业期很少考虑全方位的高可用架构，创业期属于狼性发展时段，交付效率是第一。对于京东来说也是如此，2010年京东正是疯狂快速成长时期，到目前为止数据中心规模已由原来的北京一个机房发展到了现在的多地区多个机房，由原来的单机房单中心、同城双活发展到现在的异地多活。京东做异地多活的初衷除提高灾备、高可用、提高用户体验等外，更重要的一部分是解决数据中心级别的可扩展性问题。

1. 异地多活网络架构

网络是所有互联网企业发展的根基，追溯历史，国内最初只有一家运营商，为了打破垄断，引入竞争的目的，形成了现在的中国三大运营商：电信、联通、移动。但却出现运营商间互联互通的瓶颈问题，跨网和南北互通访问速度很慢。国内的特有互联网问题经过多年的演变，虽有改善但还是存在缺陷，京东不能改变互联网的现状，但是可以根据现状优化自己的网络架构，提升系统质量，解决用户访问京东的速度和高可用问题。

网络在规划和设计时除可扩展性、冗余性外，还必须遵守一个原则，那就是越简单越好，利于后期的排障和运维。网络不同于业务系统，Bug可以无缝修复，如果核心网络架构Bug出现大面积网络瘫痪，是致命的、影响大、涉及范围广。近年伴随着Docker技术的飞速发展，目前京东线上系统已全面Docker化，网络如果还是使用物理机时代的传统网络架构，所带来的问题将是毁灭性的，异地多活的网络架构必须考虑未来的快速发展和庞大的

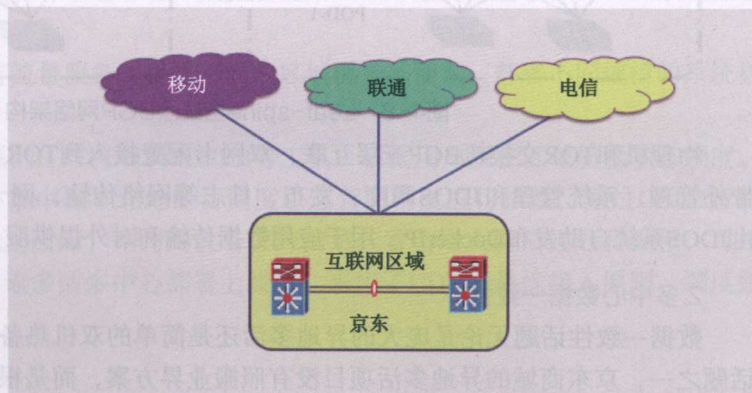


图4-7 互联网区域拓扑



Docker化集群，把网络基石做好。

京东解决运营商间互通问题使用了常用的最简单最有效的方法：把系统部署到用户最近区域以及不跨网。

京东数据中心规划和设计了互联网区域，如图4-7所示。此区域可以接入三大运营商或者其他小运营商单线和BGP线路，根据公网IP配置策略路由选择不同的出口。研发系统进行域名智能解析，GSLB调度用户访问同运营商VIP解决跨网访问。

另外还有三大运营商BGP广播、代播或直接采购二级运营商多线BGP带宽等解决方案。BGP广播和BGP带宽成本较高，大流量业务系统使用不划算，代播对于排障和failover存在很大的缺陷，这些方案在京东只适用于个别系统。比如HTTPDNS服务较为典型。

京东在数据中心内部采用Leaf-spine架构全面支持Docker自动化和发展，如图4-8所示。

- ◆ 网络设备全三层BGP互联，消灭二层烦恼。
- ◆ 服务器和TOR交换机三层互联，JDOS系统自助发布Docker明细路由。
- ◆ 每200左右机柜划分一个POD，POD之间通过边界高速互联，收敛路由表项。
- ◆ 各节点可支持横向扩容。

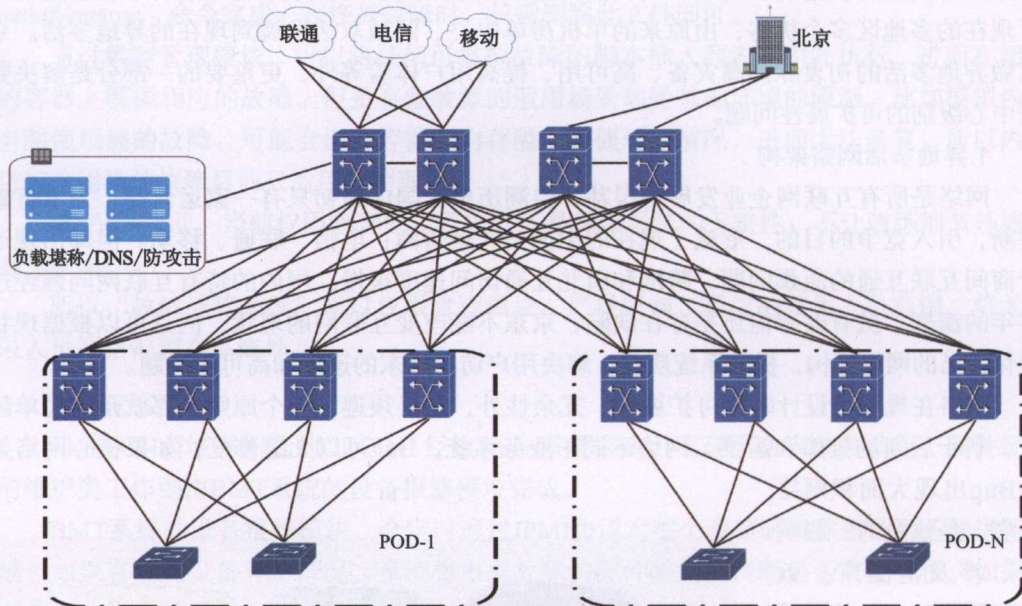


图4-8 Leaf-spine全三层BGP网络架构

物理机和TOR交换机BGP三层互联，双网卡配置接入到TOR，网卡1配置静态IP，用于带外管理、系统管理和JDOS调度、发布、日志等网络传输，网卡2和TOR遵守BGP协议，由JDOS系统自助发布DockerIP，用于应用数据传输和对外提供服务。

2.多中心数据一致性

数据一致性话题无论是庞大的异地多活还是简单的双机热备，都是所有人员最关心的话题之一，京东商城的异地多活项目没有照搬业界方案，而是根据我们现状和业务场景，走出一条不寻常之路，让研发和业务尽量透明，改动最小。实现不一样的京东商城自己的

异地多活。

对于持久化存储，根据不同的业务系统会选择不同的方案，但是最终采用的方案准则规范是一份数据最多存3份，要求至少跨两个中心的存储原则，比如一张图片或数据库，华北存2份，华南存1份。全局只有3份数据，并至少在两个中心区域以上，保证异地有备份并可用，MySQL各中心之间的复制完全依赖于京东自主研发的数据高速同步产品Tube，Tube是京东自研的基于RBR实现了Binarylog的实时采集，并且能够以管道为单位制定复制数据关系。另高速缓存JIMDB在京东内部大量系统广泛应用，JIMDB作为高速缓存通过异步持久化实现秒级同步。数据一致性主要工作在提供基础服务的基础平台部，保证业务研发不关心数据层的灾备和延迟问题。数据同步总览如图4-9所示。

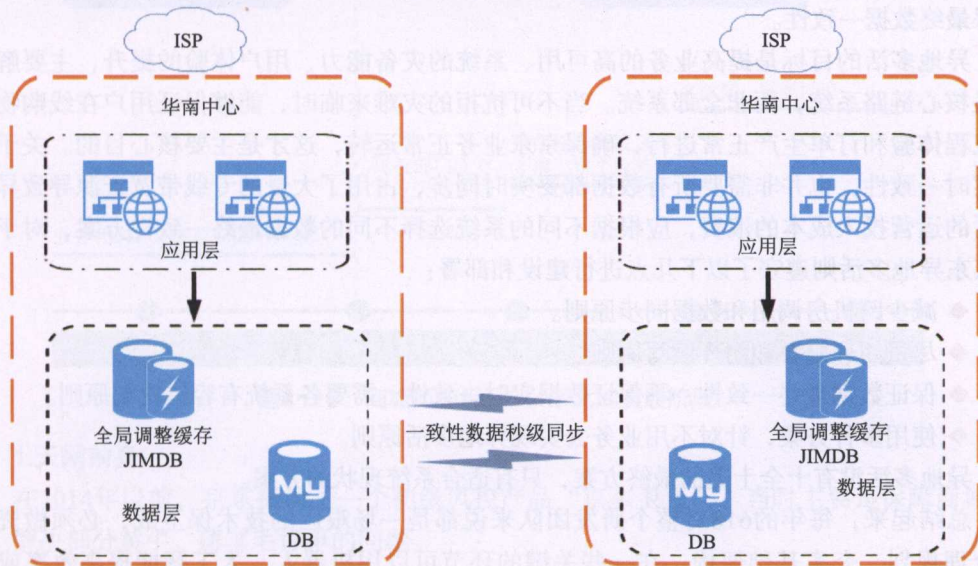


图4-9 数据同步总览

3.多中心应用部署和调度

核心应用部署闭环服务：

◆ 强依赖上下游系统闭环本地部署，数据层使用JIMDB全局高速缓存和Tube系统底层解决数据延迟同步等问题。

◆ 三大中心承担所有流量服务，根据不同的区域覆盖的面积，部署不同规模的系统集群。

◆ 调度服务使用成熟的HTTPDNS和GSLB等工具，根据用户所在区域质量动态调度。在京东发展初期数据中心集中在华北地区，调度以运营商维度进行诊断，通过GSLB智能DNS解析，如遇运营商级别故障，则跨网灾备，跨网意味着会增大网络延迟和丢包，对用户体验造成影响。现在异地多活多中心部署上线后，采用了CDN的最优接入原则，调度用户支撑灾备+多活方案。



4.多中心网络监控和灾备

多中心各种网络监控伴随着京东飞速发展已经很成熟：

◆ 各中心会有到全国省份城市的网络质量速度检测数据API，提供给GSLB作为调度数据依据之一，实时调度避开拥堵网络，选择最优中心提供服务。

◆ 各中心机房内部网络监控，包括网络设备、专线、服务器等，点到点分布式监控每个节点的网络质量。

正常状态下，各中心服务覆盖的用户，如果发生灾难性故障，无法快速恢复，如断电、断网、空调制冷故障等，则要启用异地多活应急预案快速切换，由于在线中心也是有流量和用户服务的，所以大可放心切换，不过此类切换部分系统会有损，高速缓存会及时确保最终数据一致性。

异地多活的目标是提高业务的高可用、系统的灾备能力、用户体验的提升，主要解决的是核心链路系统，而非全部系统。当不可抗拒的灾难来临时，能够保证用户在线购物核心流程体验和订单生产正常进行，确保京东业务正常运转，这才是主要核心目的。关于数据实时一致性，也并非需要所有数据都要实时同步，占用了大量的专线带宽资源导致异地多活的运营投入成本的浪费，应根据不同的系统选择不同的数据最终一致性方案，对于以上京东异地多活则遵守了以下几点进行建设和部署：

- ◆ 减少跨机房调用和数据同步原则。
- ◆ 尽量闭环服务和闭环读写原则。
- ◆ 保证数据最终一致性，不保证数据实时一致性，需要各系统有容错机制原则。
- ◆ 使用多种方案，针对不用业务去实现异地多活原则。

异地多活没有十全十美的最终方案，只有适合系统现状的方案。

总结起来，每年的618对整个研发团队来说都是一场艰巨的技术保卫战，必须做资源的合理规划，夯实基础架构。在一些关键的环节可以用机器人，人工智能提高效率或降低成本，但是最辛苦的最关键的还是要做非常充分的实施大规模的故障的演练，然后要做很认真的压测，无论是单一系统单一链路，还是全链路黄金链路都要确保无虞。每年618的整体流量都是比较大的，而且很难预期，所以还有很多的技术需要去不断地学习，不断地提高和成长。对于京东基础架构部来说，经过多年的沉淀，在既往的数据存储和计算资源之上，未来要大力投入和发展机器学习算法和AI技术应用，目的就是成为京东绝对的技术基石。



John：基础平台部中间件技术部负责人，负责JMQ、JIMDB、JFS等多个系统。

Elaine：基础平台集群部项目经理，负责京东容器集群平台项目管理工作，目前聚焦京东新一代容器引擎等研发项目。

4.2 天网系统——商城风控

在京东商城业务高速发展的同时，京东也愈来愈受到黑产链条的关注，需要应对的黑产威胁也越来越严峻，涉及面也越来越广。批量注册账号、批量抢优惠券、黄牛抢购、恶

意占库存、恶意拒收等给京东电商生态带来的影响越来越大，不仅严重破坏了商城消费者的公平性，同时也给公司品牌形象造成了极坏的负面影响，损失难以估量。

在此背景下，京东风控部门着力重点打造“天网”风控系统。在经历了多年沉淀后，“天网”目前已全面覆盖京东商城生态上百个业务节点，有效保证了消费者的用户利益。

4.2.1 发展历程

商城风控经历的4个大的发展阶段如图4-10所示。

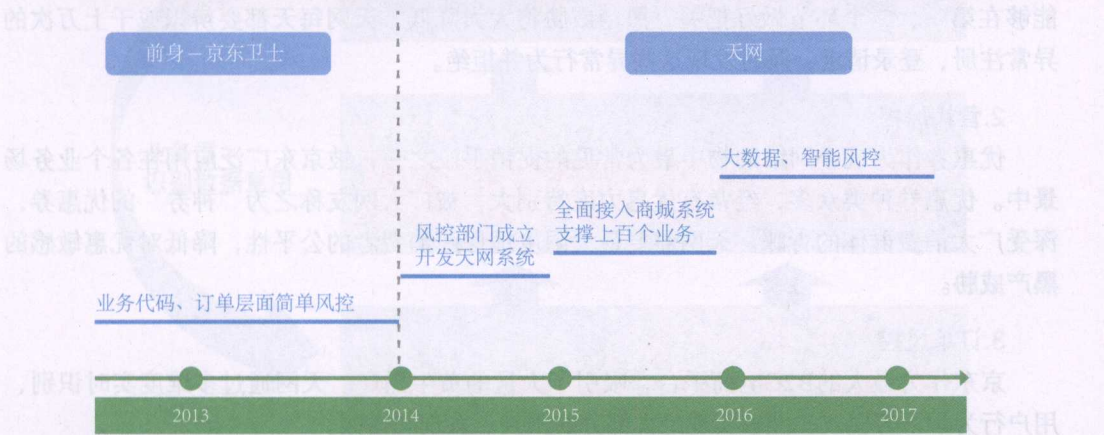


图4-10 商城风控经历的4个大的发展阶段

- 1.天网前身
在2014年以前，京东提供了一个初级风控产品“京东卫士”，当时主要是保障商城订单，解决部分黄牛、薅羊毛订单的问题。
- 2.天网成立
在2014年，京东正式成立了风控研发部门，随之推出了“天网1.0”系统，自主研发了高可用、高扩展、高并发的规则引擎，支持订单、优惠券两大维度的风控业务。
- 3.天网发力
自2015年开始，天网系统逐渐渗透到京东商城100多个业务单元，包括C端、B端，也包括国内、海外等。
- 4.天网蝶变
自2016以来，着力打造基于大数据、深度学习的“天网2.0”，以适应越来越海量的黑产威胁。

4.2.2 天网重点业务

在天网支撑的100多个业务中，最主要的有如下几个，如图4-11所示。

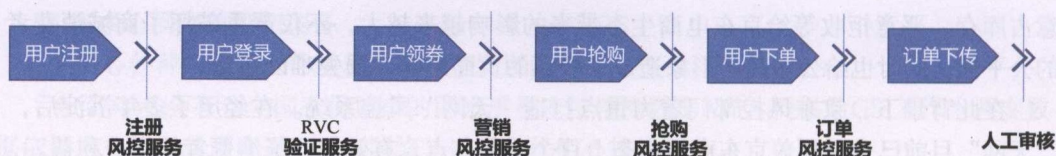


图4-11 天网支撑的重点业务

1. 注册登录风控

新用户注册，是黑产威胁的第一个关键环节，用户登录则是其第二个关键环节。如果能够在第一、二个环节做好把关，黑产威胁将大大降低。天网每天都要辨识成千上万次的异常注册、登录请求，及时发现这些异常行为并拒绝。

2. 营销风控

优惠券作为现今网络购物中最为常见的促销手段之一，被京东广泛应用在各个业务场景中。优惠券种类繁多，经常有优惠力度特别大、被广大网友称之为“神券”的优惠券，深受广大消费群体的青睐。天网需要最大限度地保障消费者的公平性，降低对优惠敏感的黑产威胁。

3. 订单风控

京东作为最大的B2C电商平台，吸引了大量的黄牛黑产。天网通过多维度实时识别、用户行为学习等模式，能够实现绝大部分的预判、发现和拦截。

4. 抢购风控

有很多一线品牌商家在京东做新品首发或是爆品抢购，这些商品的价格相对市场价格有很大的优惠力度。这就给黄牛带来了巨大的利益诱惑，他们会采用批量机器注册账号，机器抢购软件等多种形式来抢购秒杀商品，数量有限的秒杀商品往往在一瞬间被一抢而空，一般消费者却很难享受到秒杀商品的实惠。针对这样的业务场景，天网系统也能做到最大限度的风险防范。

4.2.3 天网系统架构

整个天网系统底层由三大系统组成：风险信用服务、风控数据支撑、风险监控。

1. 风险信用服务

风险信用服务是风控部门成立后第一个对外提供的服务。

2014年4月京东成立风控部门后，一直在梳理风控的业务，到5月底的时候才发现，之前一个重要的对外提供风险数据的服务由于业务增长数据量不断地增大，频频出现问题，在618时可能会对系统带来极大的风险，但是由于系统老旧，如果要整体解决这个问题，修改量太大。最后，研发团队决定重新搭建一套风控服务，并且在2周内，赶在2014年618前完成了新系统的搭建，并顺利支撑了当年的618。这套系统就是第一个版本的风控信用服务（RiskCreditService，RCS），在这套系统上又持续迭代，进行了两三次大的改进，终于发展成现在的有一百多个接入方，覆盖风控全业务的风控实时识别系统。

风险信用服务如图4-12所示，它是埋藏在各个业务系统下的风控核心引擎，既支持动态规则引擎的高效在线识别，又是打通数据和业务系统的桥梁。它是风控数据层对外提供服务的唯一途径，重要程度和性能压力不言而喻。

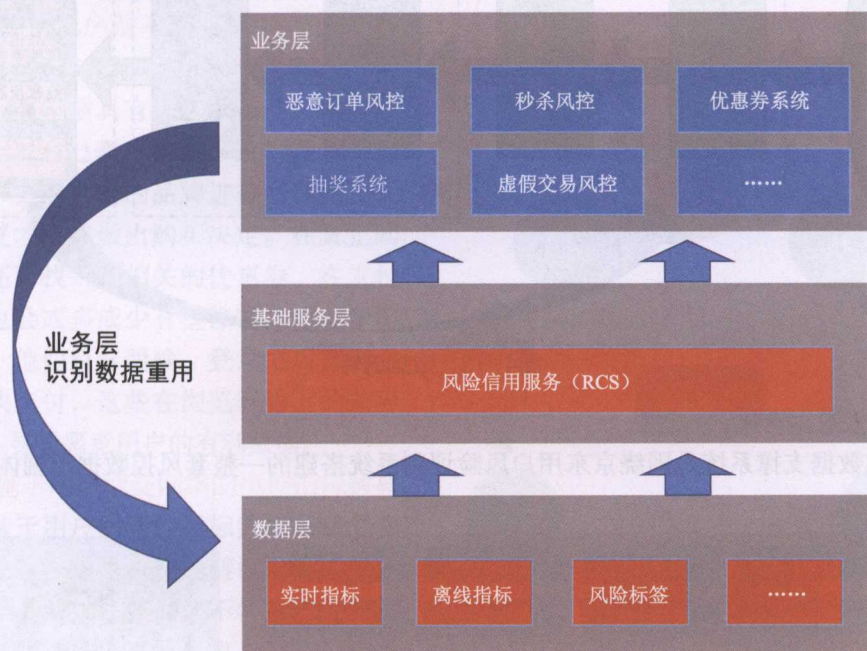


图4-12 风险信用服务

RCS的服务框架。RCS作为天网对外提供风控服务的唯一出口，其调用方式依赖于京东自主研发的服务框架JSF，它帮助RCS在分布式架构下提供高效的RPC调用、高可用的注册中心和完备的容灾特性，同时支持黑白名单、负载均衡、动态分组、动态切换等服务治理功能。

面对每秒近千万级别的调用量，RCS结合JSF的负载均衡、动态分组等功能，依据业务特性部署多个分布式集群，按分组提供服务。每个分组都做了跨机房部署，最大程度保障系统的高可用性。

RCS的数据流转步骤。风险库是RCS的核心组件，其中保存有各种维度的基础数据，如图4-13所示是整个服务体系中的基本数据流转示意图。

- ◆ 各前端业务风控系统针对各业务场景进行风险识别，其结果数据将回流至风险库用户后续离线分析及风险值判定。
- ◆ 风险库针对业务风控识别进行数据清洗，人工验证，定义并抽取风控指标数据，经过此道工序风险库的元数据可以做到基本可用。
- ◆ 后台数据挖掘工具对各来源数据，依据算法对各类数据进行权重计算，计算结果将用于后续的风险值计算。
- ◆ 风险信用服务一旦接收到风险值查询调用，将通过在JIMDB缓存云中实时读取用户的风控指标数据，结合规则模型，为各业务系统提供实时风控服务。

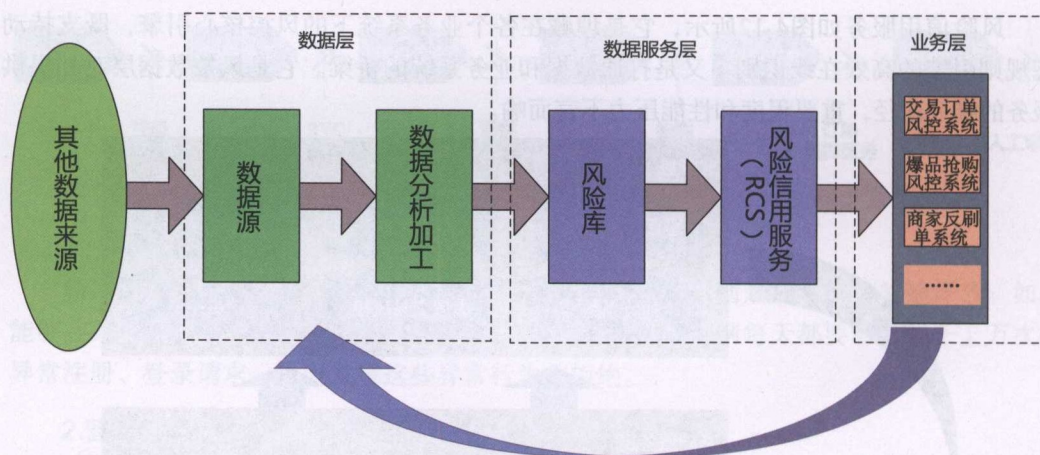


图4-13 数据流转

2. 风控数据支撑

风控数据支撑系统是围绕京东用户风险评判系统搭建的一整套风控数据挖掘体系，如图4-14所示。

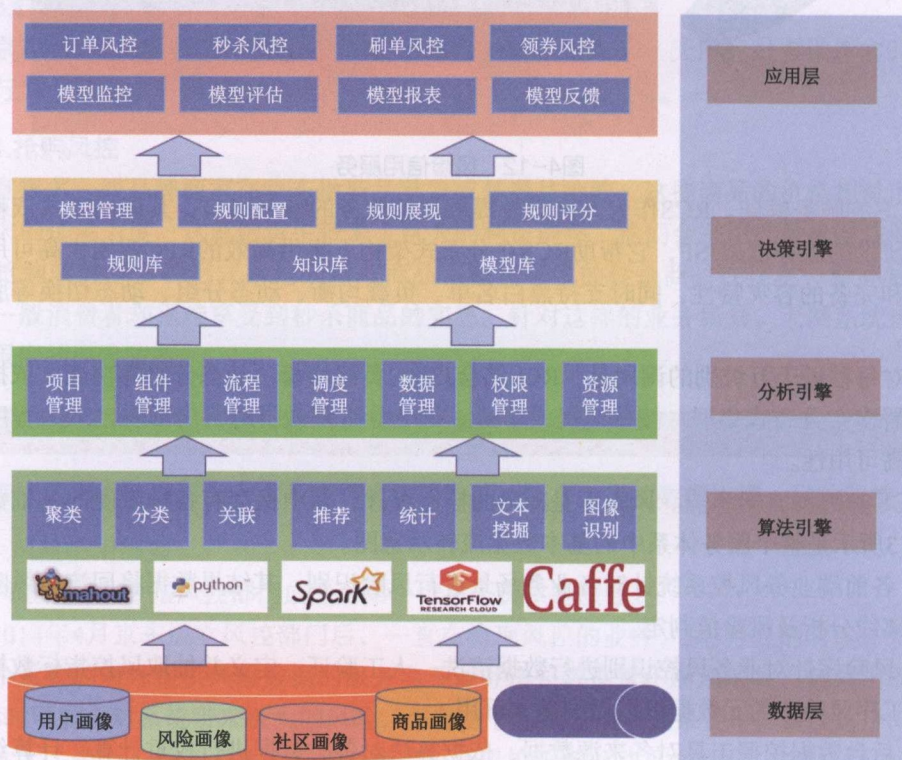


图4-14 风控数据支撑系统

数据层。数据层负责数据的抽取、清洗、预处理等工作。目前ETL程序通过JMQ、Kafka、数据集市、基础信息接口、日志接入了超过500个生产系统的业务数据，其中包括

大量的非结构化数据。通过对数据的多样性、依赖性、不稳定性进行处理，最终输出完整的、一致性的风控指标数据。这一层最关键的部分是对风控指标数据的整理。指标数据质量的好坏直接关联到系统的最终输出结果。目前指标的整理主要从以下两个维度开展：

◆ **基于用户购买流程的风控指标数据整理。**如图4-15所示，对于一般用户而言，其购买习惯具有一定的共性，例如，通常都会对自己需求的商品进行搜索，对搜索结果中感兴趣的品牌进行浏览比较，几经反复才最终做出购买决定。在真正购买之前还要找一下相关的优惠券，在支付过程中也会或多或少有些停顿。而对于黄牛来说，他们目标明确，登录之后直奔主题，爽快支付，这些在浏览行为上的差异也是京东寻找恶意用户的有利数据。

◆ **基于用户社交网络的风控指标数据整理。**基于用户网络的指标数据是建立在当前风控领域的黑色产业链已经逐渐成体系的背景下的。往往那些不怀好意的用户总会在某些特征上有所聚集，这背后也就是一家家黄牛、刷单公司，通过这种方式可以实现抓一个带出一串，抓个别找到同伙的效果。

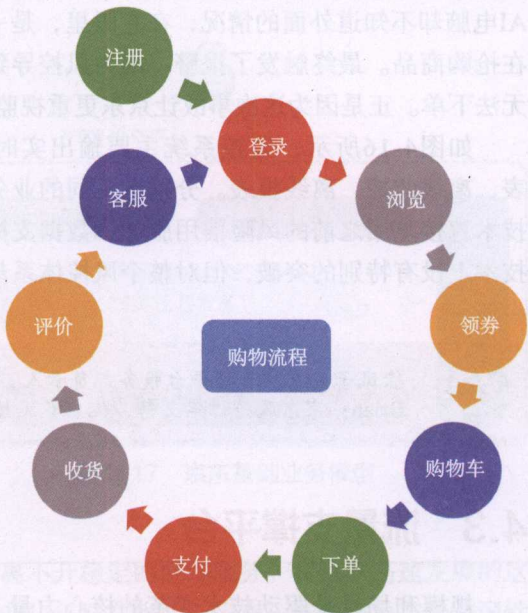


图4-15 用户购物流程

算法引擎层。算法引擎层集合了各种数据挖掘算法，在系统内被分门别类地封装成各种常用的分类、聚类、关联、推荐等算法集，提供给分析引擎层进行调用。

分析引擎层。分析引擎层是风控数据分析师工作的主要平台，数据分析师可以在分析引擎层依据业务设立项目，并在平台上开展数据挖掘全流程的工作，最终产出风控模型和识别规则。

决策引擎层。决策引擎层负责模型和规则的管理，所有系统产出的模型及规则都集合在决策引擎层进行统一管理更新。

应用层。应用层主要涵盖了决策引擎层产出模型和规则的应用场景，这里最重要的就是风险信用服务（RCS），其主要职能是对接底层数据，对外层业务风控系统提供风险识别服务。

而在模型和规则投入使用之前必须要经过风控数据分析平台，因为所有的模型和规则都在这个平台中进行评估，其输入就是所有规则和模型的产出数据，输出就是评估结果，评估结果也将反馈到决策引擎层来进行下一步的规则，模型优化。

3.风险监控系統

风险监控服务的建立背后也有一个惨痛的事故。2015年12月的时候，北京发生了严重



的雾霾。这看似不相关的事情却给风控造成了一次严重的危机：在雾霾的影响下，人们的购买行为也变得疯狂起来。大批人开始抢购口罩。原本一天只卖几百个的口罩，被消费者一箱一箱的下单，用来抵御雾霾。而待在24h恒温机房里的AI电脑却不知道外面的情况，在它眼里，是一波一波黄牛正在抢购商品。最终触发了报警，因为风控导致大量正常用户无法下单。正是因为这次事故让京东更重视监控系统。

如图4-16所示，监控系统主要输出实时监控、实时报表、离线监控、离线报表。分别有不同的业务意义。但底层技术直接复用之前的风险信用服务和数据支撑系统。虽然在技术上没有特别的突破，但对整个风控体系却是不可或缺的一环。

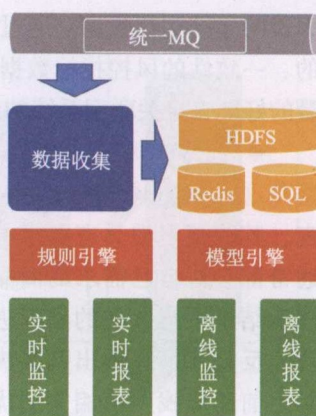


图4-16 风险监控系统



徐风子：京东风控平台服务部负责人，原京东风控研发部架构师。

Brian：京东风控订单交易、优惠券风控架构师，风控交易安全负责人。

4.3 流量支撑平台

规模和场景是驱动技术变革的核心力量。在每一年的618中，京东技术保障部都扮演着极其重要的角色，也在每一次的618备战中迭代和成长。从2003年第一次618大促到2017年，转眼已过去了14年，从最初几万的并发到目前已远超百万，我们惊喜地看到每一年618的规模都在增长，流量越来越高，同时对基础技术的挑战也日益严峻。每年在618将至的前几个月都会进入紧张的备战状态，以期上层提供更稳定的服务，给客户提供最优质的访问体验。

这8年间，京东核心网络带宽增加了10多倍，机房数量增加了近10个，服务器数量增加了数10倍，自建的CDN节点达到近百个，峰值带宽达到了TB级规模。

同时，流量和规模驱动京东进行了大量的技术变革和创新，京东的架构越来越成熟，越来越可以从容应对618场景。虽然如此，每一年618仍然谨慎有加，每一次618也都会看到服务出现这样那样的问题。在一个个无眠的长夜，研发人员守候在电脑前，紧张演练着可能出现的每一个问题，每一个可能出错的环节，那些漫漫的长夜练就了研发人员坚如磐石的技术基础和能力，从而在每一次的618酣战中大获全胜。

4.3.1 稳定压倒一切

在每年的618备战会上，都反复强调稳定的重要性，稳定是整个618的首要前提。而这两个字对技术保障部的各个业务有更重的分量，技术保障部支撑京东所有上层业务，为京东商城、京东金融和京东云提供整体的基础网络、统一流量接入、CDN加速和防攻击服务，是业务稳定的基础，也是用户体验的基础，责任重大。而每一次618，全站流量都会毫

无疑地创造历史，因此，每一次618都是一个崭新的命题，对京东的系统和技术提出更高的挑战。

2009年以前，京东的业务规模相比今天真的是沧海一粟，当时所有的业务加在一起也没有太多流量。整个电商业务的玩法也比较简单，那时候图书业务还不是京东的大战先锋，还没有京东金融和白条，没有京东超市和生鲜，也没有提前消费。因此，每天京东的流量都是富有规律地潮起潮落，一切都是幸福的模样。随着京东业务的迅猛发展，对基础服务的依赖也日益增加，特别是一年一度的618，对底层服务的考验是历史级的。用户访问京东的每一个请求都会经过前端接入，CDN加速，京东网络的复杂交互，以及防攻击系统，最终到达上层业务处理后返回，其模型如图4-17所示。

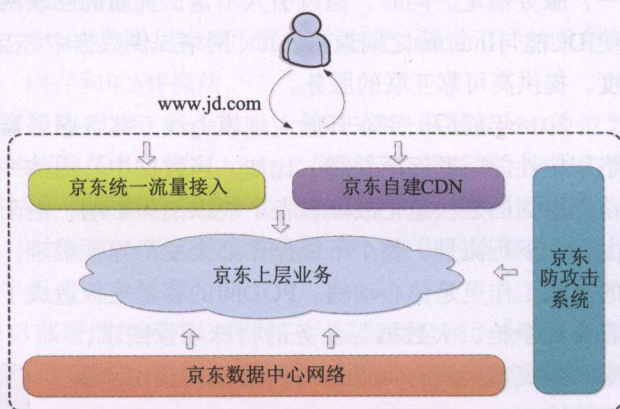


图4-17 京东基础业务模型

网络是互联网的基础，一切优质的应用都离不开稳定的网络服务。在京东高速发展的这些年，京东网络建设一直走在业务的前端，至2017年已在华北、华东、华南建立布局全国性的核心机房，同时第一个自建的数据中心已在华东上线，内网互联带宽达到百GB规模，为京东发展提供透明优质的网络服务。

同时构建了京东全局流量接入平台，用户每一个请求都会从京东LB（前端负载均衡）接入，统一进行用户安全验证、权限控制和负载均衡，并提供四层和七层互联，为业务交互提供保障。

为进一步保证用户访问质量，京东自2009年起便开始自建CDN的筹备，目前全国范围内已遍布上百个节点，接入带宽达到TB规模，为用户提供高质量的内容加速服务。

京东自研的防攻击系统，实时监测机房流量，并对异常流量进行清洗，为用户的访问保驾护航。

这些服务组成了京东最基本的网络基础设施，为上层产品和业务提供可靠的服务。作为基础技术部门，构建高可用的底层服务始终是京东的终极目标。618就像是一年一度的大考，检验着团队努力的结果和成效。

4.3.2 基础网络

1. 网络架构的升级和扩容

京东网络团队为整个京东集团提供基础网络服务，各个功能模块解耦，可以分为三类内容：



互联网数据中心（IDC）内部：服务器等主机设备的网络接入。

IDC出口：数据中心的互联网接入服务。

IDC之间：数据中心之间的互联服务。

IDC内部网络互联所有网络设备和服务器，实现机房内无阻塞的高速转发，结构统一，服务稳定。同时，通过引入丰富、优质的互联网资源，将数据中心接入公网骨干层，使IDC能与Internet之间通信。IDC网络提供数据中心互联，实现数据中心之间流量的灵活调度，提供高可靠互联的服务。

2015年前后，京东开始大规模上线万兆级服务器，随着数据中心规模的增长，旧的网络架构性能问题逐渐暴露，比如，当数据中心内的一台核心交换机出现故障时，运维人员首先想到的是快速把故障设备下线以消除影响，由于设备间的横向带宽不足以支撑所有的上行、下行流量，整个下线操作都会变得异常艰难，稍有不慎就会造成重大事故，修复后的上线工作更是惊心动魄。POD间的容量短板造成了频繁的扩容，数据中心内业务间存在着资源争抢，大数据等业务等特殊场景需求，未知的流量突发等问题都对网络架构提出了新的挑战。

2017年，基础网络团队设计了全新网络架构，新架构实现了POD内1:1带宽收敛比，达成了数据中心内的无阻塞通信。新的京东三层网络架构如图4-18所示。

新架构采用全三层结构，放弃了大量的虚拟化技术，每台网络设备属于独立的逻辑单位，可以方便地对任意一台网络设备进行上/下线操作；取消同层设备间的互联线路，解决了故障时设备间横穿流量过大的问题；统一的路由协议规避了路由重新分发时，引起路由环路的风险；消除了原有数据中心的大二层网络不稳定因素，使得新架构下的网络更加健壮稳定，并在2017年的618大促中发挥了重要作用。

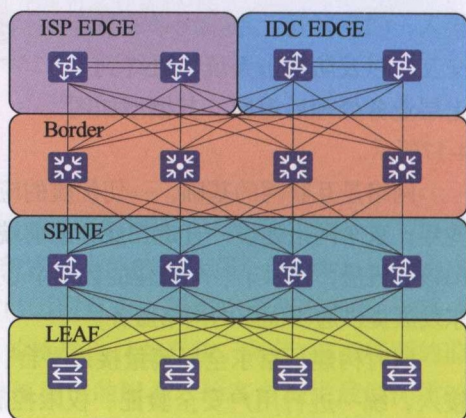


图4-18 京东三层网络架构

2. 网络自动化

近年来随着公司业务的发展，网络团队面对的日常事务也成倍增长。因此，如何提高效率解决线上操作，成为了最迫切的问题。

每次在有业务部门报故障时，网络团队都需要消耗大量人力去进行故障排查，整体效率低下；发生网络问题时，无法通过监控迅速定位具体的影响范围，导致排查故障的过程异常艰辛，如同大海捞针。同时，网络团队每天还会接收大量的业务变更需求，都需要消耗大量人力来处理此类事务，这种状态如果不快速改变，随着规模的逐渐扩大，整个团队将会被这种重复性事务拖垮，工程师将没有精力去思考更有价值的事情。

监控和基础数据是自动化的前提，如果无法解决这些问题，自动化便无从谈起。业务系统在发生故障时，优秀的监控系统可以确保工作人员在第一时间发现问题，提高工程师

的排查效率，协助定位问题，从而快速恢复业务。同时，自动化系统可以大幅降低人为失误的风险，并提高效率。当监控工具和自动化工具有机结合时，就可实现真正的自动化运维。

网络团队通过自主研发基于SNMP、ICMP、Netconf等协议的监控平台Joyeye，实现了针对网络自身的白盒监控和模拟用户流量的黑盒监控。

白盒监控，对全网所有设备进行100%的监控，针对不同种类的端口制定不同的采集策略，从端口流量、CRC到设备的CPU、内存和ICMP探活。

黑盒监控，将大量业务部署到监控客户端（Agent），中央模块根据Agent的位置（Rack、POD、机房）为其下发不同的探测任务，同时Agent之间进行交叉探测，并将监测到的数据上报给中央模块，这样可以站在业务侧的角度，更加真实客观地反映出网络的健康状态。通过各种细颗粒度的监测，中央模块支持按服务进行汇聚，获得网络服务各种维度的健康状况，如机架、集群、POD、机房、专线等。

对于公网部分，在所有机房的出口部署监控程序，针对三大运营商全国所有省市的探测节点进行探测，通过这部分数据可以得知数据中心出口的网络健康状态和各地运营商的网络健康状态。

监控系统提供了很多有价值的基础数据，而对这些数据的加工也一样有价值。通过自主研发的网络自动化平台Joypaw，实现监控之外更多的高级功能。例如，某业务部门需要为10台服务器配置Trunk功能，自动化工具可以通过底层基础数据得出IP地址和设备端口的映射关系，针对设备端口进行配置，并更新CMDB。Joypaw通过“CLI+Netconf”对接网络设备，针对各厂商的接口进行适配，在用户接口和设备接口之间增加中间层，屏蔽掉各个厂商的接口差异（CLI和Netconf），并提供统一的北向接口供业务方调用，提升部署效率的同时，也避免了人为操作的失误。开始时，每个功能模块都是零散的单一功能，后续可以将多种单一功能通过排列组合的方式组成复杂功能，根据日常事务消耗精力的比重和自动化的投入产出比，按照不同的优先级逐个开发新的功能。Joypaw平台模型如图4-19所示。

当监控系统和自动化工具都趋于成熟，自动化运维平台也会孕育而出。在监控系统中，加入日常的运维逻辑，在报警的同时加入单个或多个处理动作，最终实现人力资源的解脱。例如，通过公网的监控，可以得出每个运营商出口、每个机房出口的健康状况，当达到告警阈值时，在告警的同时，加入运营商故障流量切换的动作，可以大大降低服务切换时间。

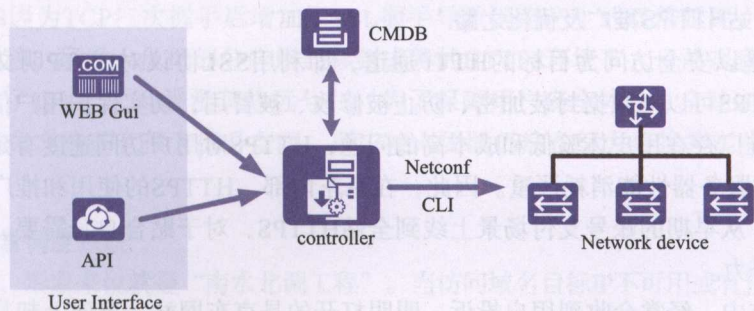


图4-19 Joypaw平台模型



4.3.3 京东高可用前端接入系统JFE

京东用户接入团队承担了全站业务流量接入工作，全站请求都先到达前端接入，经过防盗链、安全审核、SSL握手等一系列规则后，才代理到业务集群。用户接入流量聚合到一个部门管理，不仅更好地收敛用户流量，而且各高可用产品以及衍生周边产品集成到平台化管理，京东高可用前端接入系统JFE（Jingdong Front End）就是这样应运而生的。JFE系统包含了京东自主研发的高性能四层和七层负载均衡、全局流量调度系统GTC（Global Traffic Control）、运维平台管理等，针对618超高并发场景实现了前端系统稳定、透明转发、智能灾备，其中多项技术已成为公司标准。

本节主要介绍京东全站HTTPS以及全局流量调度系统GTC，图4-20是京东用户接入系统架构，由域名系统、高可用JFE、流量调度系统及监控系统组成。

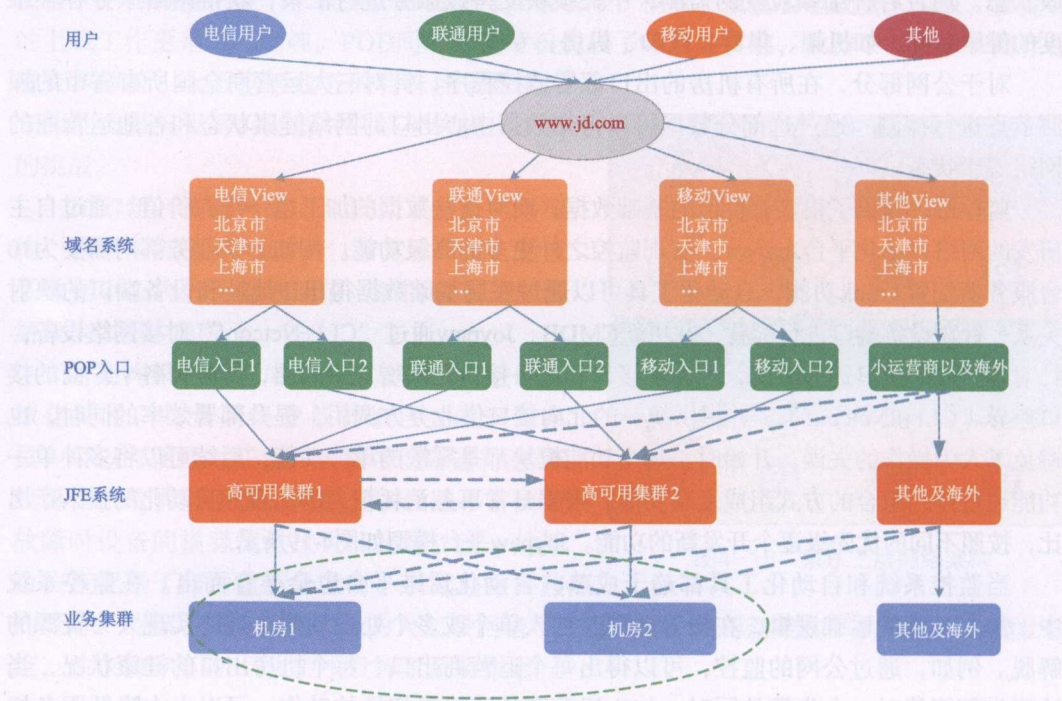


图4-20 京东用户接入系统架构

1. 京东全站HTTPS推广及优化之路

HTTPS是以安全访问为目标的HTTP通道，即利用SSL协议对HTTP明文数据加密传输，使用HTTPS可以对数据封装加密、防止被修改、被冒用，切实保护用户信息。但是早前HTTPS的推广存在用户体验低和成本高的问题，HTTPS对用户访问速度有延时影响，同时SSL计算对服务器性能消耗严重。因此，在京东内部，HTTPS的使用和推广也经历了很长一段历程，从早期的账号支付场景上线到全站HTTPS，对于贴合自身需要，京东做了很多的优化和努力。

日常工作中，经常会收到用户投诉，明明打开的是京东网站，页面上却是其他内容；

或者总弹出其他不相关的广告，这两种都是典型的网站内容劫持。而对电商网站来说，页面内容又直接关乎用户信任感和订单量。

用户请求自客户端发出至到达京东服务器之前，有着很长的链路，在明文传输的过程中，网站内容很容易被劫持和篡改，严重影响用户体验，也会对京东本身造成损失。而要改善这种问题，上线HTTPS对传输内容加密，几乎是唯一方案。而众所周知，全站HTTPS会带来证书、服务器、技术研发、服务改造等一系列成本。特别对618场景，全站HTTPS会有很大的压力风险，每一个请求都需要进行SSL握手和内容加密计算。但是在用户体验为先的前提下，一切问题都不再是困难，团队经过了近一年调优和改造，实现了全站HTTPS，并在2017年618期间稳定服务。

全站HTTPS的先决条件。独立站点启用HTTPS的经典步骤：购买证书、配置证书、修改路径、测试、服务上线。而全站HTTPS上线则完全不是线性增加的复制粘贴，京东全站HTTPS作为集团战略项目，从立项、上线、完成、评估、验收到经历618考验，几乎长达一年时间。

那么京东如何落实全站HTTPS呢？京东所有流量都由JFE统一接入，这给用户接入团队执行全站HTTPS提供了统筹安排、全局配置的基础条件。前端接入系统有统一的运维管控平台，管理着用户接入京东的所有软件版本、VIP自动分配、配置文件、证书管理及流量监控及调度、预发布系统等模块，这些模块一起配合完成了京东全站HTTPS上线。其中预发系统、HTTP/HTTPS双支持、硬件加速等问题是其中比较重要的几个环节。

京东全站HTTPS上线之后，外部解决了用户购物体验，内部用户接入团队也迎来了更大的技术挑战，全站HTTPS是否能扛住618流量峰值考验，一直是大家担心的问题。从项目初期上线到稳定应对618流量洪峰，用户接入研发团队从多方面对HTTPS进行了优化和改造。

◆ **SSL证书改造**。由于是京东统一前端接入，所有域名都会解析到固定的IP，即这些IP会承载所有京域名流量，这给HTTPS的上线造成了一定困难。因为上千个域名都需要经过这些IP接入，每个IP就需要加载多张SSL证书，以适配不同的域名请求。SNI是一贯的解决方案，但由于部分客户端并不支持SNI，导致用户无法适配到正确的SSL证书以致HTTPS失败。为了更好地适配，用户接入团队摒弃了早期的部署方法，上线SAN框架，可以多域名签发一张证书，完美解决了这个问题。

◆ **性能优化**。京东全站HTTPS解决了用户安全访问问题，但也牺牲了一定的用户访问速度。HTTPS因为TCP三次握手后增加了SSL握手导致网络产生约双倍延迟，为根本解决此协议弊端问题，研发人员把部分应用迁移到成熟的HTTP2环境下，目前观察HTTPS应用访问性能提升了15%。“路漫漫其修远兮”，为了保障用户安全体验，全站HTTPS只是开始，HTTPS优化之路还有很大提升空间，用户接入团队仍然在努力寻找并实践着更佳解决方案。

2. 全局流量调度系统

流量调度，形象来说就是“南水北调工程”。当访问域名目标IP不可用或有预期问题时，通过域名解析或其他手段调度到正常IP地址，以达到正常访问网站目的，即流量调度。举个例



子：

- ◆ 联通用户访问网站正常解析到联通IP地址；
- ◆ 但联通IP地址不通；
- ◆ DNS解析强制把域名解析到正常的电信IP地址。

京东全局流量调度模型如图4-21所示，当某多线机房联通入口故障时，快速干预域名解析结果，将流量由电信入口接入，实现流量调度。

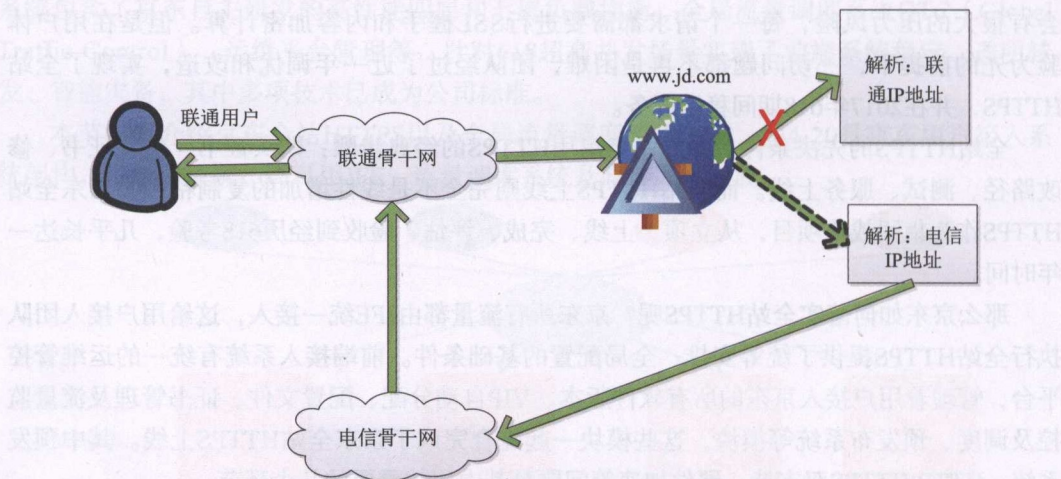


图4-21 京东全局流量调度模型

全局流量调度系统的诞生。用户接入平台是客户端用户流量经过的第一道关卡，畅通无阻才可以分发到各产品线服务器进行处理。无论日常运维还是618期间，团队的核心工作就是保障系统“稳定性”。京东有上千个域名，多个机房、多个运营商入口，复杂的架构下如何快速恢复业务是研发团队的目标，流量调度系统应运而生。

当前京东的用户接入系统架构，从用户请求到服务器响应，用户的流量经过“域名系统”完成解析到VIP，VIP通过“用户接入系统”完成流量分发到各服务器。京东当前DNS域名视图已精确到省级运营商颗粒度，也就是京东可以精确到对省级运营商用户进行流量调度。国内南北方已建设多个物理机房，每个物理机房除具备本地运营商接入外，还可通过网络专线接入异地机房运营商，这样机房就具备了多个本地或者异地运营商入口。在跨机房集群间组建了BGPD+LVS负载均衡集群，机房同时发布所有VIP，并通过路由策略控制VIP路由转发。这样的多网点接入为用户流量调度系统提供了基础条件。

全局流量调度系统的实现。用户在域名系统、用户接入系统具备流量调度的条件下，如何根据故障场景从多个维度定义流量调度策略，原则上遵循公网用户同运营商调度，以下选择最常见的两个场景：

- ◆ **机房流量入口故障。**机房故障或者升级系统需要迁移流量。

定位故障机房：定位到某机房故障，例如A机房。

获取故障机房VIP：管控系统已记录所有VIP信息，包括机房、运营商、区域运营商等，筛选遍历到所有故障机房VIP。

定义目标机房：根据故障入口定义到可切换的目标入口，例如B机房。

获取故障域名及解析VIP：通过故障源VIP匹配到域名，并记录域名解析的VIP；

目标VIP：通过域名解析的VIP检索匹配到目标机房IP，故障VIP除外，优先选择目标机房IP，此根据优先级选择其他机房入口。

更换解析：更新所有视图的解析记录，生效权威DNS配置。

◆ **运营商流量入口故障**。某机房本地或者异地流量入口故障。系统整体实现过程如同机房故障，差异之处源故障VIP：对所有机房接入此运营商VIP聚合。目标VIP：同运营商优先选择跨运营商。

智能流量调度系统。为解决切换的时效性问题，快速恢复业务流量，京东依托健全的监控平台自动定位到机房、运营商流量入口故障点，并根据预置的目标机房、目标流量入口完成自动切换，目前流量调度系统已常态化在线运行。

4.3.4 京东CDN

在京东技术发展史上，CDN是一个非常典型的场景驱动型业务。在最初用户规模和流量都不算大的时候，京东曾长久依赖商用CDN提供的内容加速服务。随着后来流量越来越大，特别618带来的规模和稳定性需求，依赖第三方的短板日渐凸显。因而，在2009年开始了漫长的CDN自建之路，至2017年已过去了8年，这8年刚好是伴随京东618成长的8年。2017年京东自建CDN已可百分百地满足业务需求，并随着京东云的日益发展，CDN技术已经输出至公有云，给越来越多的用户提供优质的内容加速服务。

1. 自建之路

2009年，京东第一个自建CDN节点在天津机房上线，标志着京东高可靠CDN的建设正式拉开序幕。翻开那一年的历史，虽已尘封，但仍旧惊心动魄。每一项技术的变革和创新也都伴随着具体的历史场景和需要。

某天，团队接到大量用户投诉，西北多省用户无法正常访问京东，经过排查，定位到是CDN的问题。和当时的厂商反馈，又经过漫长的等待和排查，对方确认是线上存在错误配置，导致用户无法正常访问。但由于是和其他用户共享CDN集群，回滚和生效流程复杂，那个故障自发现至全部恢复耗时超过3h。

这对京东和用户来说都是无法接受的。而同时，京东商城对CDN的依赖也在日益增加，随着业务多样性的增加，越来越多的内容需要通过CDN加速。而类似上述的情况依旧层出不穷，于是，自建CDN的概念就在团队脑海里逐渐清晰起来。

2009年京东正式组建了CDN团队，从零基础进入，到出色完成京东618保障，自建CDN在致力于提高京东用户体验方面正发挥着越来越重要的作用。就在刚刚过去的2017年618已有近百个节点在线提供服务，并且在这一次实现了618期间全站HTTPS，在用户访问质量和稳定性方面都有着卓越表现。回首来看，正是当初的问题驱动京东做出了正确的决定，而今随着618规模越来越大，业务越来越丰富，对CDN的需求也越来越严苛，而自建CDN无疑是正确的唯一之路。



京东CDN发展历程如图4-22所示。



图4-22 京东CDN发展历程

每一个阶段都伴随着京东业务的发展，和上层服务对CDN的诉求，每一个阶段都有着其特点和技术变化。直到今日，CDN业务依旧在迅猛发展着，方兴未艾的直播和视频业务使得CDN有着更为重要的角色，而京东自建CDN在技术的道路上也越走越好。

图4-23是京东CDN多级缓存模型，当用户打开京东首页时，会首先去最优的CDN边缘节点加载文件，如果边缘节点丢失，则回源到二级节点，如果再次丢失则会回源。至2017年，两层缓存可以保证系统90%以上的缓存命中率，而覆盖大区的三级缓存也已在规划之中。

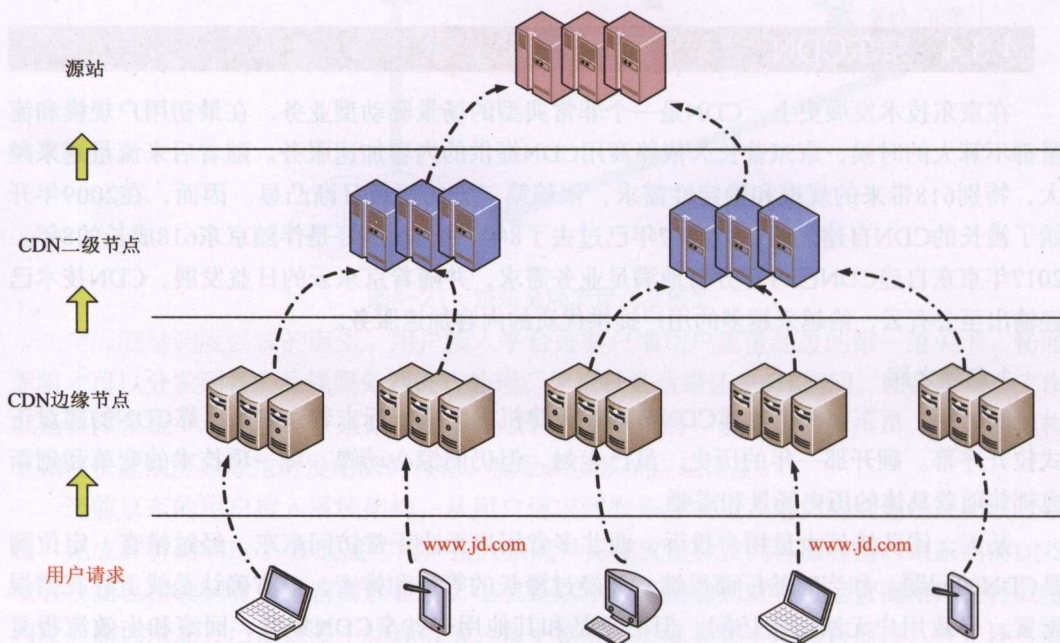


图4-23 京东CDN多级缓存模型

随着京东业务的迅猛发展，CDN带宽需求也日益增加，从最初的几GB带宽到十几GB，再到几十GB，CDN质量对京东的用户体验至关重要。而京东自建CDN从一开始就是为满足618高并发而进行的，研发团队使用了一系列高可用架构和技术，保障服务足够的冗余和稳定。最近一次618中，京东的峰值带宽接近TB，而CDN整体稳定可靠。

2. 节点升级，保障618稳定可靠

京东有着非常丰富的业务形态，包括京东商城、京东金融、京东云等，每一个又包含了千千万万的子类和业务，而每一个都离不开CDN加速。每逢618，CDN团队的压力都特别大。根据历史规律，这8年来，每一次大促CDN业务的流量都在增长，且峰值会达到平常的数倍。因而，每一次我们都在见证历史。

在历史级的流量面前，是如何应对用户流量和自身容量的矛盾呢？

◆ **硬件升级，实现全站万兆网卡改造。**在京东第1个节点上线时，标配的千兆网卡，在那个时期无疑是够用的。随着单节点流量越来越大，千兆网络已不能满足带宽需求。在2017年完成了全站万兆接入改造，为节点扩容提供了硬件基础。单节点容量提高了10倍。

◆ **软件升级，提高缓存命中率。**在京东CDN日趋成熟的过程中，软件架构也在不断升级。近年京东实现了全站Squid向ATS的迁移，更高的性能、更灵活的扩展性带来了更多的产品feature和性能。通过调度整合，内容压缩优化，研发团队实现了更高的存储比，单节点缓存命中率稳定在90%以上。

◆ **业务分离，实现性能最大化。**缓存里的文件有类型和大小之分，根据不同文件类型使用不同的存储方案，如针对小文件上线SSD集群，提高单机I/O性能；对于大文件则采用SATA集群，提高单机容量，以存储更多内容，最终实现资源合理分配，最大化提高CDN性能。

◆ **引入商用CDN，加强服务保障。**在京东CDN的发展史上，一直都伴随着商用CDN的影子。从最初的完全依赖商用，到自建初期，再到成熟期，商用CDN一直都在为京东提供着稳定的服务。从可用性的角度看，多一家厂商分担流量，系统整体故障的风险就会下降。因而，在自建技术完全成熟的今天，仍然在使用商用系统。

而在非618期间，京东自身流量开始下降，剩余带宽则通过公有云提供给其他用户，既保证了618性能，又保证了资源利用率。

4.3.5 防攻击系统

随着618的逐渐临近，京东的每条业务线都为这即将到来的购物狂欢而努力着，旧的业务需要扩容，新的业务需要上线，越来越大的流量规模和越来越复杂的业务场景是每次618京东防攻击系统最大的挑战。

技术保障部的防攻击团队主要负责的是网络攻击的防护，特别对于DDoS，从1996年有记录的第一次DDoS攻击以来，这种成本低廉、简单高效的攻击手段就长期困扰着每一个经营线上业务的企业。不仅因为DDoS攻击防御对网络、性能资源要求高，同时随着攻击手段的不断进化，攻击者也伪装地越来越像真实用户，大量的攻击混合在正常业务流量中，业务和安全的权衡是DDoS防护的一大难点。

1. 物理资源

DDoS攻击的规模每年都以指数级增长，想要对抗这种巨型的攻击，首先需要在硬件设施上达到足以对抗的规模。京东在数年前就已完成异地多机房、多线路的网络布局，核心业务在京东的每个骨干节点中都会部署热备服务，同时依靠用户接入部维护的智能调度系统进行负载均衡，当任意一个节点可用率下降时，通过智能调度系统便可以将业务流量收敛到正常节点中，保证业务稳定性不受损失。

同时每个节点都会常态化保持70%以上的链路冗余，用冗余的带宽来对抗随时可能到来的DDoS攻击流量，冗余的带宽越大，成本越高，安全系数也越高，如何在控制成本的同时，保证足够的安全系数，是防攻击系统的一大任务，特别需要考虑在618期间，正常业务流量增



长迅猛，会大大压榨常态的冗余链路，所以需要进行数据分析，进行机房带宽的扩容。

京东对稳定性的要求极高，需要保证90%以上的攻击能在不影响业务的情况下进行透明防御，剩余10%里有90%的攻击可以通过流量调度解决。在这个目标之下，对硬件环境、网络带宽及架构都是要求极高的。

2. 网络结构

DDoS攻击通常是针对4层协议的攻击，通过制造巨型流量耗尽目标业务带宽或服务器资源。防攻击系统需要在攻击流量进入业务集群之前进行清洗，保证业务集群能够正常稳定运行。

京东防攻击系统，简称JDTP（Jing Dong Traffic Protect System）。如图4-24所示，JDTP直连边界交换机，在业务网络中部署2个子系统，JDTP-Detec和JDTP-Clean，前者负责监听京东IDC所有的进出口流量情况，由边界交换机将与ISP通信的流量通过镜像发往JDTP-Detec；JDTP-Clean负责对攻击流量进行清洗，并将正常流量送回业务网络之中。通过与边界交换机建立BGP邻居，将需要清洗的IP引流至JDTP-Clean集群中处理，利用BGP协议实现ECMP，达到负载均衡、高可用的效果。

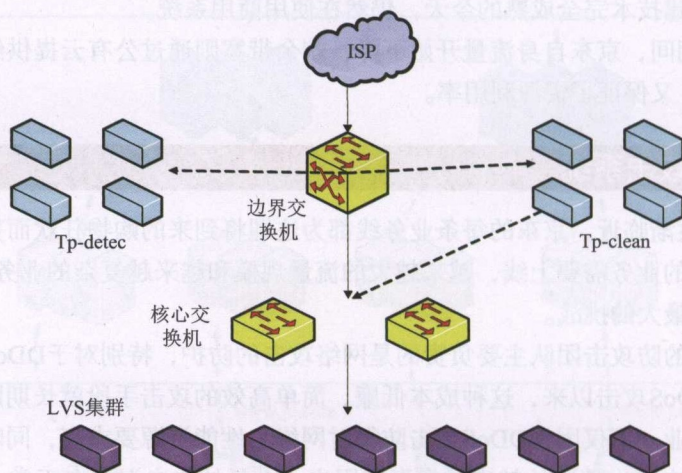


图4-24 JDTP系统网络部署图

3. 软件架构

传统的DDoS攻击检测是靠简单的阈值判断来进行的，通过统计某个周期内发现的数包数目，与配置的攻击阈值进行比较，超过阈值的就是攻击行为，小于阈值的就没有攻击。现代DDoS攻击的复杂度不断增强，攻击维度越发复杂化，报文特征随意变换，传统的防御方案越来越难以适应现在的攻击场景。

京东内部的业务十分复杂，首页、搜索框、购物车、POP、秒杀、优惠券等，每个业务都有自己的流量特征，同时类似直播等新兴服务也在不断上线，每个业务在618期间的流量都很难预估，没有办法配置合适的防御阈值，因此需要在原先的道路上开辟出一条新的防御体系。

新的防攻击分为两大核心：一是攻击发现系统，二是攻击清洗系统，DDoS攻击的

清洗策略经过多年发展已经非常成熟，大部分的攻击如SYNFLOOD、UDPFLOOD、DNSFLOOD都能够基于协议特性进行拦截，但由于大部分的清洗策略对正常流量来说都无法做到完全无损，因此怎样做到只对异常流量进行清洗，让正常业务流量无损地流到业务端是优化整体防御效果的重点。新的防御架构中以攻击发现系统为核心，将攻击事件聚焦在最小的流量特征中，将对整体流量的影响减到最轻。

为了在618庞大的业务流量中发现攻击事件，攻击发现系统被划分为4个层级，如图4-25所示。

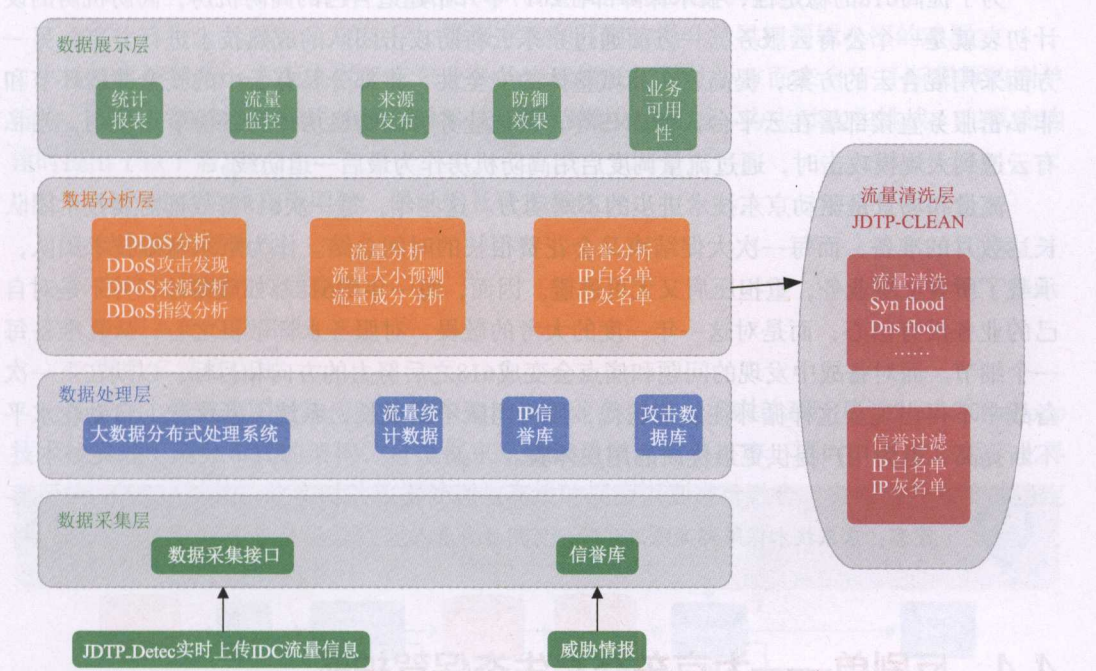


图4-25 JDTP系统架构图

◆ **数据采集层。**要发现攻击首先需要对线上的流量做到100%监控，利用部署在边界的JDTP-Detec系统，对机房镜像流量进行全量的统计，统计的维度可以根据分析策略进行定制化，整个数据采集层在618时的处理峰值达到了100GB的流量处理能力，同时多达10万条的秒级统计数据被送往攻击发现系统的数据处理层。

◆ **数据处理层。**使用Kafka进行数据分发，数据处理层除实时流量数据外，还存放着攻击历史数据、IP信誉数据等发行系统所需数据，数据分析层的各分析模块将根据需求来获取数据仓库中的数据进行攻击分析。

◆ **数据分析层。**这是攻击发现系统的核心模块，任何对攻击事件处理有益的分析模块都在这一层中，利用数据处理层的攻击数据和机器学习方法，对流量的规模、成分、访问者的行为特征、攻击的指纹数据都进行全方位的分析，学习这种对维度的分析体系，才是对抗现代DDoS攻击的必要手段。

◆ **数据展示层。**为了运维和应急响应能够更加及时和高效，可视化也是非常重要的一步，通过运维后台，值班人员能够快速地发现攻击事件，并分析防御手段是否生效，让攻



击在造成影响之前，被及时处理。

4.混合云方案

随着云服务的逐渐发展，越来越多的基础服务开始采用云方案，同时云安全的概念也逐渐得到重视。云服务本质上来说也是一种共享经济，比如DDoS防御，云安全厂商通过售卖自己的冗余带宽和技术，将购买带宽的成本分摊到所有使用云服务的用户身上，云服务用户则可以付出大大低于传统服务的成本购买到更加安全的环境。

为了提高618的稳定性，技术保障部在2017年开始建造自己的高防机房，高防机房的设计初衷就是一个公有云服务，一方面通过京东云将防攻击团队的成熟技术进行分享，另一方面采用混合云的方案，提高京东商城整体的安全性，将部分私有云中的安全薄弱环节和非私密服务直接部署在云平台，如DNS等，其他业务将高防机房作为备份节点使用，当私有云遭大规模攻击时，通过流量调度启用高防机房作为最后一道防线。

流量和场景是驱动京东技术进步的不竭动力。这些年，每一次618备战都需要技术团队长达数月的准备，而每一次大促结束又会花费很长的时间总结。作为京东基础技术团队，承载了所有上层业务，重担压肩又无可逃避。因而，每一年的618都如履薄冰，并不是对自己的业务没有信心，而是对这一年一度的大考的敬畏，对服务永怀敬畏之心，认真准备每一个细节。而对备战中发现的问题和痛点会变成618之后努力的方向和目标，以期在下次备战中不再出现。这样循环往复地迭代，使得团队不断成长，系统不断优化，自动化水平不断提高，也为用户提供更最优质的用户体验。



霞客：京东技术保障部高级工程师，可用性负责人。

4.4 反刷单——为京东交易生态保驾护航

随着电商竞争的愈演愈烈，商家对于流量的渴求越来越强烈，虽然京东提供了广告、秒杀、活动专场等营销工具来帮助商家获取流量，但仍有部分商家不满足于此，采取一些不正当方式来获取流量来源，其中，刷单就是一种典型代表。这种行为会极大扰乱京东的交易生态：对平台而言，会带来虚假繁荣，不利于正确预估平台的发展速度；对商家而言，优质商家利益会受到损害，最终劣币驱逐良币；对用户而言，将难以甄选购买到真正需要的商品，从而降低对于平台的黏性。因此，京东对于刷单采取的是“零容忍”的态度，除了严格的管理和奖惩规则，更是利用先进的技术手段，通过大数据形成一个完整可靠的反刷单技术屏障，维护好京东的交易生态，保障用户的最大利益。

在目前的刷单生态中，刷单已经成为一种赚钱营销的工具，渗透到各类综合、垂直电商与O2O平台，并日趋规模化、市场化及产业化。目前，市场已涌现出各种刷单公司、集团、平台，“从业人员”众多，“产业”日趋成熟。刷单公司为商家提供一站式服务，提供“保质保量”的服务保证，而且刷单手法日趋隐蔽、逼真、多变，刷手们通过多种刷单工具和软件来提高刷单效率并隐藏真实身份，更有甚者，雇用大量的人肉刷单者，刷单行

为尽量模拟真实用户行为，隐瞒刷单意图。这些都给刷单识别带来了严峻的挑战，也激励着团队不断优化和改进系统来对抗越来越隐蔽的刷单行为。

4.4.1 反刷单系统架构

在搭建一个完备的反刷单系统的过程中，会面临到以下重大挑战：

- ◆ 多维度的数据引证。刷单行为的异常通常会分布在多个维度的数据中。
- ◆ 敏捷的模型迭代。刷单行为变化快，需要模型快速迭代响应。
- ◆ 快速识别，准确与召回提升。模型准确和召回的提升是风控领域永恒的难题。

这些系统挑战一方面来自于京东业务的快速增长，另外一方面来自于日渐猖獗的刷单行为。因此，构建业内领先的反刷单系统是亟待解决的问题。为应对这些挑战，在系统构建时提出了以下需求：

- ◆ 分布式系统的高可用性、低延迟及可扩展性。
- ◆ 结果的可复现性。
- ◆ 多样化数据的可适应性。
- ◆ 决策系统的灵活性。
- ◆ 服务多维度应用。

基于以上需求，同时为应对京东快速增长的业务规模，反刷单系统基于Spark和Hadoop技术栈实现了纯分布式的架构，可根据京东的流量增长实现扩容。同时，系统采用了业界流行的LAMDA结构，将实时的数据处理与离线的模型识别进行融合，大幅提升系统的稳定性以及对于各种特殊业务场景数据的支持。系统架构如图4-26所示。

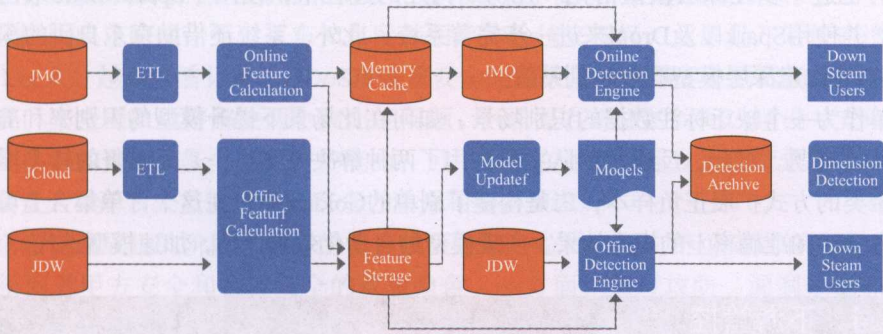


图4-26 反刷单系统架构

4.4.2 特征工厂

1. 京东交易链条的完整数据及其特点

反刷单系统使用了京东交易链条的完整数据。这些数据具有以下特点：生命周期长，从用户产生消费冲动到对商品发表评论，一个订单关联到的数据跨度可长达数周甚至数月；数据种类多，包括日志、买卖方属性、商品属性、交易属性、支付、物流等；数据变动频繁，在订单生命周期内交易数据都可能发生状态改变。



2. 数据预处理功能

为完善地应对和处理这些数据，在更长的时间跨度上，从海量持续变动的数据中挖掘刷单行为的痕迹，系统设计了完善的数据预处理功能，包括：

- ◆ 基于Hive和Pig的批量数据处理框架。
- ◆ 基于Camus和SparkStreaming的流式数据处理框架。
- ◆ 基于Oozie的业务调度系统。

3. 特征工厂框架

同时，还设计了一套完整的特征工厂框架以便简化特征开发过程，规范特征管理。该框架包括以下三个部分：

- ◆ 使用特征工厂的离线初级特征。
- ◆ 基于SparkMLlib以及自研开发的离线高级特征提取。
- ◆ 基于SparkStreaming实现的实时特征提取。

这些框架上线后大幅缩减了新版特征的开发时间，在特征量增长200%的情况下缩短了特征提取时间的80%，为后续高效准确的刷单识别打下了良好的基础。

4.4.3 决策引擎

特征工厂框架的提取对于用户的下单行为已经有了系统化的描述，在此基础上，还需要构建一套决策引擎从中识别出刷单订单。

在决策引擎的构建过程当中，系统从最初的规则模型演变到了目前线上使用的浅层模型，同时也进一步往深层模型衍化。在初期构建的过程中，使用了SparkMLlib来构建最初的模型，并使用Spark以及Drool来进一步完善系统。此外，系统还借助京东自研的深度学习平台PINO来构建深层模型刷单识别系统。

刷单作为一个缺乏标注数据的识别场景，如何在此场景下提升模型的识别率和准确率是业界公认的难题。对此，反刷单团队主要采用了两种解决方案：一是无监督的样本增强，通过样本聚类的方式扩展正负样本；二是构建了刷单的GoldenSet，在这个订单集合上可以计算出模型在召回和准确率上的提升效果，实现模型的自动化效果评估，加速模型迭代。

4.4.4 下游管控

反刷单系统只是京东交易生态管控中的一环。为打造更好的京东交易生态，反刷单必须与下游系统联动管理。目前的刷单管控主要包括：联合搜索团队，将刷单识别结果用于搜索排序权重；联合评论团队管控刷单评论；联合京东商城管控刷单店铺以及商品的营销资源排位；联合奖惩团队进行刷单店铺的奖惩管控；联合京东金融进行金融征信管理；联合业务风控团队进行账号风险等级管控。

在后续的业务发展中，反刷单数据也将接入更多的下游使用方，力争将刷单数据覆盖京东交易系统中的每一个环节。同时将刷单数据推向上游，联合前端团队，管控账号注册及登录行为，以期对京东账号体系实现更全面、有效的管控。



榆木脑袋：数据专家，擅长机器学习及风控管理。

4.5 信息安全——重于泰山

交易是京东商城最核心的业务流程，提高京东商城用户的体验，保障用户安全、放心的购物是京东集团信息安全部的主要职能之一。

京东业务经过10多年的快速发展，信息安全的能力建设进入了高速发展时期，业务模式从“补丁—应急”为核心的救火模式向“防范—感知—管控”为核心的自适应模式。目前京东集团信息安全团队覆盖了京东商城的各个业务部门，业务涉及保护业务连续性、用户及企业数据、账号业务安全、保护用户利益、云安全、IoT/智能家居、AI安全等诸多安全领域。

同时，信息安全部门与集团公关、法务、监察、客服等多个部门建立了联动机制，“集中管理、统一指挥、步调一致，协同保障”，为京东的业务安全保驾护航。

4.5.1 筑起安全的长城

1. 快速适应——伴随业务需求构建安全能力

在京东内部，安全团队称2011年为“京东安全元年”。在这一年，京东集团高层将过去分散在各个业务部门的安全职能汇聚在一起，成立了专职的信息安全团队。

安全团队成立的这一年，京东知名度和用户体验越来越好，业务量开始快速增长。同时，SQL注入、越权查看订单、DDoS攻击等安全问题数量在内部安全测试和外部的报告中频繁出现。安全团队亟待解决两个突出的问题：一个是在最短的时间内建设一支可靠的安全技术团队，构建Web安全基础防护能力；另一个是在业务快速发展的背景下，如何让业务团队负责人和研发人员快速补充安全方法论和安全知识。

在京东快速成长的阶段，业务的快速成长是第一位的，具有不可挑战的权威，安全团队同时扮演者甲方安全和乙方安全的双重角色。一方面，通过攻防、漏洞挖掘和自研工具，把存在的安全风险告知业务部门，为业务部门提供更透明的安全服务；另一方面，安全团队也在借鉴乙方安全公司的安全理论和实践经验，补足业务上的安全短板。

2. 体系对抗——安全产品和架构

安全的攻防不再是个人或团队的技术对抗，而是体系的对抗。安全也不应该是后加的内容，而是在产品或服务构建初期就嵌入其中，就像一架飞机的每一个零件都必须是安全可用的。

和传统企业安全说的“三分技术、七分管理”不同，京东安全强调技术层面，使用“七分技术”来保障决策层的策略能够落地。安全团队从Web业务安全、办公网和IDC安全以及安全管理三个层面进行了分阶段、有重点的建设。整体架构如图4-27所示。



图4-27 京东安全整体架构

京东电商业务系统具有“频繁迭代、敏捷开发、快速上线”的特点，商业化的安全检查和扫描产品，对于业务而言都过重，扫描和检查时间无法满足要求。为此，安全团队根据业务特点，采用“取其精华”的拿来主义，开发出个性化的扫描脚本，不但提高了准确度，而且效率也能够满足业务发展的严格要求。

同时，安全团队以开源的Nmap、OpenVAS扫描器作为重要的技术补充手段，构建了整体的Web安全扫描体系。通过对扫描和监控进行了优化和自研，已经做到一天内完成对全网数十万台服务器和应用系统的安全检查。

京东安全的SDL也非常具有特色，它覆盖培训、需求、设计、评估、发布、监控、响应、改进等一系列环节组成。SDL最早由微软提出，在国内一直水土不服，但SDL在京东绝不是一个空泛的流程框架，不是简单的几个安全扫描检查点。京东SDL与业内执行的SDL最大区别在于体检模式的不同。

第一，安全体检后自主选择改善目标。通过对各部门所用的CI系统、项目管理系统、上线系统、漏洞管理系统、人资系统等系统的分析，得出各团队执行SDL前的安全体检值。业

务部门自行根据自身的发展现状，自行选择希望达到的安全体检值。有了目标之后，信息安全全部有针对性地为业务部门导入所需的SDL阶段和活动，并指导业务部门如何提升。

第二、宽进严出。信息安全部在每一个阶段都制定了考核标准，比如，在安全测试阶段，信息安全部为部门提供对应的培训，每个团队完成培训后必须定期反馈结果。没有完成考核的部门不予通过，没有通过考核的部门需要重做本阶段的活动，直至指标达到要求。

第三、“授人以渔”而非“授人以鱼”。通过SDL的推行，将信息安全部的能力输出给各业务部门。POP团队通过执行SDL，在部门内部建立了测试的标准化和自动化的工具，保证部门内的所有项目都统一执行标准，不但自行测试出项目中的安全漏洞，而且自行发现隐藏在现有系统中的安全漏洞。通过SDL的“授人以渔”，信息安全部做京东的安全御林军，SDL、金蚂蚁就是京东的安全护卫队。

3.殊途同归——为了用户的微笑

每年的618安全保障都是安全团队工作的重中之重，不能有丝毫的闪失。每年的大促期间会有很多的风险，不但有可能出现安全漏洞，而且会出现劫持、钓鱼、攻击、欺诈等很多的情况，必须依靠整体的力量。

京东安全的大促重保以安全委员会为首脑，围绕重保安全指挥中心，在技术管理团队、业务管理团队和支撑管理团队的联动和配合下，共同完成保障任务。每年的重保，安全团队都会用一张巨大的纸，列出备战和保障的各个事件类型、主负责人、各部门对接人，以及手机号、邮箱。任务分工明确、职责清晰，极大提高了事件响应的效率。618保障的整体架构如图4-28所示。

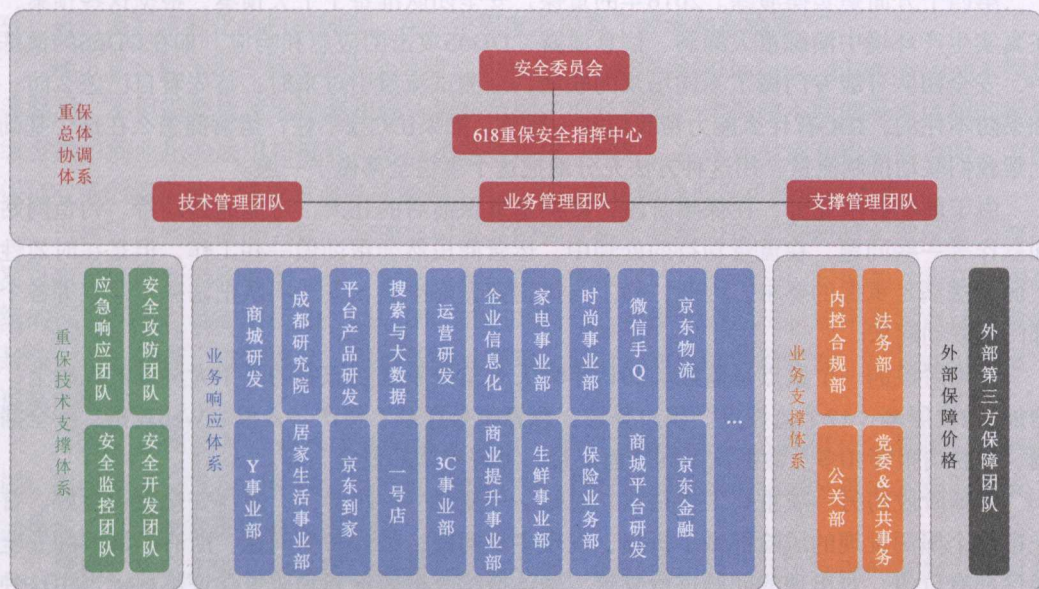


图4-28 618保障整体架构

安全保障框架包括两层：总体协调层和技术保障层。总体协调层负责协调各个部门的负责人，统一响应事件。技术保障层在指挥中心的统一调度下，依据相关的职责，步调一



致地将保证动作做到位。

从时间上分，618的保障主要分为备战阶段和保障阶段。在备战阶段，安全团队从以下几个层面进行备战。

第一个方面是安全风险。在每一次重保的备战时，把京东针对大促期间存在的风险都会列出来，做差距的比较。包括引导发布、资产分析、隐患自查、响应演练、监控感知。为了保证比较高效地完成跨部门的协调工作，引入了引导发布的工作机制，通过引导发布机制让所有人都能够知道每个层面应该如何加强安全，安全团队首先自己会把所有涉及不同层面的东西来复盘，制定好之后让各个业务部门自查。

第二个方面是资产分析。重保备战前一个半月，安全团队集中搜集各个业务部门的核心业务系统，各业务线安全官负责对业务系统的系统架构、网络通信、数据存储、业务逻辑、账号权限、系统部署配置等安全控制点进行检查，对系统进行源代码级别的安全检查和渗透测试，将核心业务系统的安全风险降低至可接受的水平。截至2017年618大促活动，信息安全团队累计投入10000多人日，检查了3000多个核心系统，避免了大量可能发生的损失。

第三个方面是JSRC。每年的重保之前的一个多月，京东安全响应中心每周都有一些不同的活动，2016年618保障时最大的奖项是无人车，通过重奖，保证重保之前把所有的风险都发现出来。在2017年618大促，信息安全部发动JSRC的1000多名白帽子，开展了“全民保障618活动”，从最北端的哈尔滨到最南端的海南岛，从最东端的上海到最西端的新疆，各省市、地区的白帽子，在618之前积极告知网络中安全漏洞，累计提交漏洞217个，其中深层次、影响范围大的高危严重漏洞20余个，在618大促前消除了隐藏的安全漏洞。

第四个方面是实操演练。2016年的重保，安全团队准备了十大预案，根据这些预案，在真实生产环境中演练重大漏洞、信息泄露、DDoS攻击的应急和响应。如在DDoS的演练中，安全团队分级专门做了不同场景的联动，当测试流量引过来时，首先看自己怎么防；如果防不住的，IDC有什么能力帮我们扛一部分；如果IDC扛不住，运营商怎么在这个基础上帮我们联动抵制流量。用这种方法充分地把这个事情全部做了一遍。

由于前期备战充分，在保障阶段，特别是重保前后的几天，可能遇到劫持、钓鱼网站和欺诈等一些问题，在流量劫持的处理中，运营商虽然会帮助做一些工作，但是在时效性上面无法满足重保的要求。因此，除了与运营商做好准备，安全团队把法务、监察等各个部门也联动起来。

保障阶段的一个重点是监控。2017年哮天犬监控系统在原有的一个大屏幕的基础上，增加了2块屏幕，12个监控点。用户登录、恶意账号登录、资产、入侵、Web攻击、安全漏洞、威胁情报等30多类数据从不同层面汇聚，呈现整体的安全态势。

保障阶段的另一项重要工作是从“总结——复盘——改善”，使得安全变得更好。首先把整个漏洞发现的风险都做了折损，发现多少问题，处理多少问题，所有的漏洞和安全事件都有价值。对出现的问题进行复盘，确认持续改进的计划和监督保证落地。当用这种方式展示保障带来的收益和避免的损失时，得到了公司的认可。

4.展望未来——寻找下一个战略高地

在未来，京东安全希望重新思考安全问题的根源本质，构建最高水准的有效的安全能力。

在自研和能力积累层面，京东安全在人工智能和机器学习在入侵和感知、安全漏洞扫描、移动应用安全、IoT安全的应用，在无人机、无人车的身份认证、业务安全与风险控制等方面提升京东的技术水准和用户体验。

在技术合作层面，京东与行业伙伴在技术和联盟上展开紧密合作，提升安全能力。例如，与英特尔共同推进先进技术在电商平台上的落地应用，与腾讯在防诈骗等层面进行共享数据信息，联防联控等。

另外，黑产的对抗和打击也是京东安全一直在努力的方向。安全团队与内控合规等部门一道，联合政府、行业和执法部门共同打击网络黑产，让网络黑产无处遁形，为用户打造可信赖的网络环境。

在行业人才的培养方面，希望与全国的各大高校一起设立相关的信息安全课程，让学生可以拥有黑客的安全视角，从人才储备的源头上就建立良好的安全能力。

4.5.2 来自攻防第一线的博弈

1. 全面覆盖——快速响应的技术挑战

京东仿佛一艘巨大的宇宙飞船，在快速向前飞奔，安全团队就像这艘飞船中永不休息的机械工程师，如果出现了安全问题，就对它进行修缮，保证这艘船能够安全地到达目的地。

最初，安全团队使用标准的扫描器，对全网进行通用型的扫描。不但扫描时间长，而且存在一定的误报和漏报问题。遇到的第一个问题是商业化通用扫描器的速度和准确度问题，攻防团队和应急响应团队通过对漏洞的研究和分析，按照扫描的优先级和类型进行区分，自己编写定制化的扫描脚本，在一定程度上提高了效率和准确度。

接下来，对于那些特定的安全风险，比如2017年初爆发的Hadoop、ElasticSearch未授权等漏洞，不但需要快速地检测和确认影响范围，而且需要一个跨越不同的隔离区域的轻量级方案，商业化的通用扫描器无法满足这些需求。同时，在保证发现和检测能力时，怎样保证业务的稳定性，降低对业务的影响，将不可控的风险降低至可接受的水平。

当建设这些扫描能力后，发现如何统一的调度各个扫描引擎和资源时，满足各个部门对京东全网的扫描和检测工作。包括：解决海量中快速的应用识别、扫描和检测在应用系统的深度问题、日志解析和过滤、URL的归并问题、数据状态同步、通用漏洞审计、数据封装、消息处理、进程调度、海量数据处理等。同时，对系统的稳定性、可靠性也提出了更高的要求。

另外，如何让其他业务团队和相关人员了解到总体的安全态势和安全漏洞的影响范围，感受到安全发现和检测对自己切身的影响，把扫描和检测的结果从被动的获取变成主动推送，也是非常重要的一件事情。

现在，从京东安全团队成立到现在，业务部门的安全已经具备了标准的防御力。但是面临着一些特别的威胁：

◆ **全局性的突发重大安全漏洞。**电商业务需要业务系统具备快速水平扩展的能力，这要求所依赖的软硬件环境保证高度的一致性。同时，避免重复造轮子，信息系统底层广泛



使用开源的框架和组件。在这样的场景下，一旦某个开源组织底层的架构存在安全风险，导致整个系统的某一个点被突破，对业务造成的影响都是非常巨大的。这要求安全团队必须从全局出发，全面了解各个信息节点的安全态势和行业的威胁情报，时刻监控这些安全风险趋势，在安全漏洞大规模爆发前，快速完成修复。

◆ **数据安全**。用户数据是京东的最为核心的资产之一，是京东的“首要机密”，一旦被窃取，将会是巨大的灾难。特别是安全与业务相冲突时，如市场部做活动，很多时候会提取一些用户的数据出来做抽奖，服务需要请求姓名、电话、邮箱等完整的信息，如果打码，活动无法进行，如果不打码，一旦疏忽就可能导致信息泄露。这要求安全团队必须在数据产生、数据使用授权、数据安全传输、数据安全交换和使用、数据加密存储和安全销毁过程的各个环节进行管理和控制。

2. 技术+自动化——秒级响应的自信

京东安全的威胁管理以技术和自动化为核心，整体架构如图4-29所示。业务部门所属的应用系统及其IP地址、端口、服务和组件信息同步到大海资产管理系统中，扫描引擎定期从大海系统中增加扫描对象并加载扫描任务，并提交到扫描引擎。业务部门的源代码管理系统也与扫描引擎自动化对接。一旦扫描完成，将扫描结果发送至脉象平台，用于整体安全态势的分析。一旦发现安全漏洞，则通过ITSM工单系统进行跟踪，安全漏洞的跟踪和修复状态定期通过安全数据周报的形式告知各业务部门。

另外，安全官检查和外部接报的信息安全事件，经过验证后，一方面通过ITSM工单系统进行跟踪；另一方面，在大海系统中检查存在问题的关联系统和信息，调度引擎再次执行关联检查和扫描，杜绝安全漏洞的次生影响。

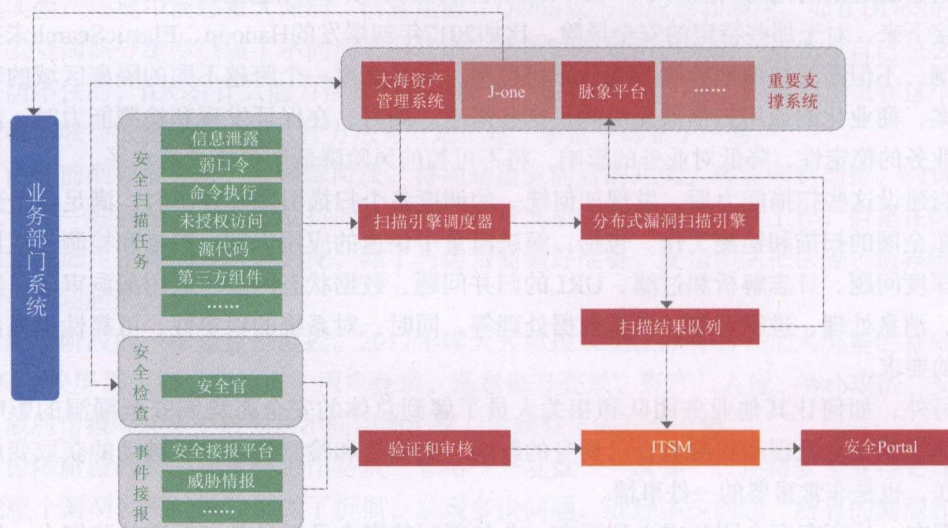


图4-29 安全漏洞闭环管理

对于不同级别的安全漏洞有不同的修复时限的要求：严重漏洞1天，高危漏洞3天，中危漏洞7天，低危漏洞14天。

若漏洞仅是被修复，则有可能导致其他次生的安全风险，不能从根本上解决问题。为

此，安全团队建立了重大漏洞的跟踪及改进流程。在高危漏洞应急结束后的3个工作日内完成漏洞的分析和评审，并确认改进措施，由安全官监督和验证改进措施的落实情况。高危漏洞跟踪及改进流程如图4-30所示。

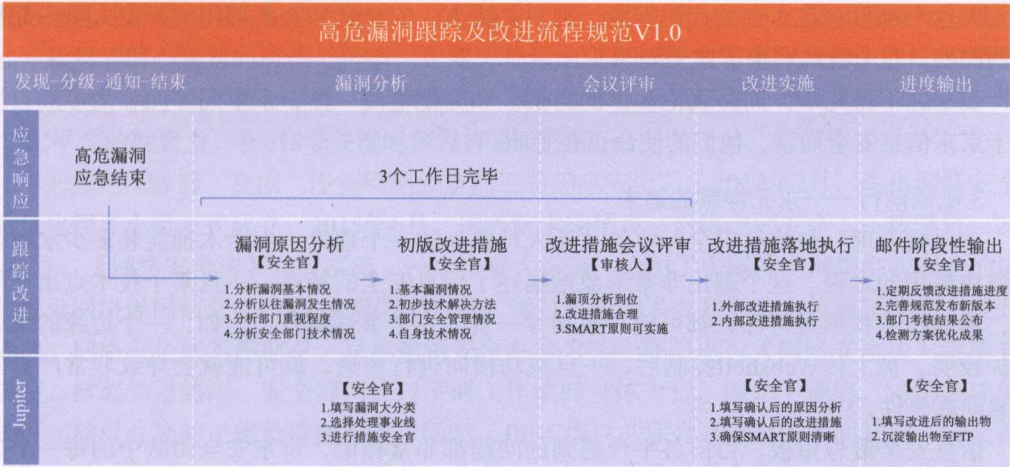


图4-30 高危漏洞跟踪及改进流程

在源代码扫描的安全检查层面，是通过扫描程序源代码中的漏洞代码特征，进行针对性的漏洞查找。在前期的使用中，也遇到了很多困难，主要有如下四方面。

- ◆ 源代码分布较为分散。上百个研发团队分管自己的源代码版本控制系统。
- ◆ 代码库地址、域名及研发负责人的ERP没有被对应起来，浪费了大量的运维精力。
- ◆ 扫描引擎默认扫描规则检测效果不理想
- ◆ 扫描引擎默认界面只能手工提交扫描任务，难以批量扫描，效率低下。

因此，京东源代码审计系统从京东上线平台（J-one）同步数据，解决源代码管理分散的问题，通过Web Services接口实现自动提交扫描任务。

在最关键的扫描规则上，根据京东多数应用系统使用SpringMVC+Mybatis框架，使用Maven管理和构建项目的业务特点，设计并研发了文件上传、SQL注入、通用组件漏洞等规则。如Spring MVC使用MultipartHttpServletRequest或MultipartFile作为方法参数或使用javax.servlet.http.Part作为方法参数，进行源代码检查规则的编写。

在业务链条中安全使用数据方面，对数据进行高强度加密的同时，对请求的频率进行了安全限制。对一些必须提取用户数据的场景，会给这些信息附加一些“key值”，有了这个动作，就可以记录这些信息的使用时间、使用量级，还有负责人。而且这个key带有有效期。例如：一个月的过期时间，过期之后就不能读出用户信息了。这样，哪怕是出现了信息泄露，也可以追踪溯源，查到责任人。

尽管总体上安全无虞，但安全团队日常活动仍然需要协调各部门修补无数的小漏洞。例如，2016年5月份，安全团队在业务系统Web服务器端发现了一个低危的跨站漏洞，协调开发部门修复问题后没有意识到这个问题在移动App端同样存在。直到从外部接报，移动端同样爆出了问题，才完成了修复。



2017年618之前，安全团队把京东核心业务域名都进行了一次大排查，针对研发人员经常犯的错误以及严重漏洞逐一排查。在2017年5月初，统一收集了各业务部门的系统后，攻防人员手工检测一个新业务时发现了一个注入点，有可能窃取数据。经过和业务沟通，最后发现这个项目存在六七处代码问题，那只是周五，安全官和业务部门的研发人员一起加班到深夜，在上线前解决了这个问题。

安全远不是儿戏，而是真枪实弹的对抗。和其他电商一样，京东同样面临诸多战役。对于京东信息安全而言，他们的使命正在于此。

3. 砥砺前行——永无停歇的追求

“木桶原理”是信息安全行业的从业人员熟知的一个理论，一个木桶能装多少水取决于那块最短的木板。这个理论非常形象地描述了信息安全的特点，一旦某个技术点出现短板，其他技术积累得再好，都可能导致功亏一篑的安全事件爆发。比如，一个边缘的系统防护较弱，被上传Webshell控制后，一旦成功横向进行渗透，就可能就会导致非常严重的信息泄露事件。

信息安全最忌短板，目前各平台遇到的问题都非常相似，京东安全团队中的每一个人也都在探索中，包括如何完善自己的防御体系。这包括如下几个方面：

- ◆ **更加强化的入侵感知体系。**包括对网络恶意流量的清洗、网络层的入侵检测、主机层的入侵检测、Webshell的检测、数据库的入侵检测、统一的应用层快速防护能力等。

- ◆ **办公网的安全保障。**包括BYOD的安全防护、WiFi的安全防补、终端补丁下发的状态检测和下发结果检查、APT攻击的发现和检测、敏感信息的泄露等。

- ◆ **更坚实的传统安全防御措施。**包括规划整体的安全业务区域，构造纵深防御体系，对现有的服务器、端口、组件、应用的安全配置管理和加固。

- ◆ **移动应用的安全防御和加固。**包括移动App的安全检测、安全加固、统一签名管理、移动安全沙盒、设备指纹、分发渠道的监控和管理等。

- ◆ **业务风险控制和隐私保护。**包括账号的风险控制、多设备的统一登录管理、恶意订单、打击羊毛党、打击黑产、反爬和反欺诈以及个人数据的保护等。

4.5.3 全民保障618

2013年对于信息安全行业来说是不寻常的一年，这一年，各大互联网公司开始意识到安全构建需要内部的强大加外部的配合，如此才能更好地知晓并抵御威胁。2013年4月，京东安全应急响应中心应运而生，作为安全团队对外的窗口，成为企业与白帽子间沟通桥梁和情感纽带，肩负了京东安全品牌和形象建设的重要使命。

1. 披荆斩棘——JSRC成长与运营

JSRC的建立体现了京东对待安全问题的重视程度以及对外开放交流的心态。四年多的时间里，平台上已拥有1000多名白帽子，每年提交数千个漏洞及情报。平时，JSRC平台白帽子会通过威胁信息和情报的搜集，帮助京东提早感知威胁，做到安全问题的预防、判断和解除。每逢618等大事件期间，平台白帽子、安全厂商会协助京东安全部门，更严谨、更

及时地响应安全风险，协调处理安全威胁，共同保卫用户安全及企业安全。从防微杜渐到逐步建立铜墙壁垒，这是大家共同的目标和方向。

有人说SRC的成长过程是“三分建设，七分运营”，因此JSRC力求以最快的感知安全威胁速度并及时响应为根本立足点，以打造良好安全圈生态为目标，整合资源及力量，通过一系列的精心策划的活动与传播，定期向外输出京东的技术安全实力并保持JSRC运营工作的稳定、有序、持续地进行，不断推陈出新，给平台白帽子及合作伙伴带来惊喜，增强JSRC平台的凝聚力，在618等关键时刻发挥重要作用。

把握时代脉搏，贯彻“没有网络安全就没有国家安全”。2014年2月，中央网络安全和信息化领导小组正式成立；2015年6月，国务院学委增设“网络空间安全”一级学科；2017年6月，《网络安全法》正式实施，一个个重大举措，说明了中国捍卫网络空间安全的决心和建设网络强国信念。JSRC紧跟国家号召，通过自身的平台和影响力传播正能量，提升引导力。网络安全法实施前夕，京东邀请安全专家为整个部门进行《网络安全法》专项知识讲座，掀起学习热潮。安全部门经过平时工作中的身体力行，深化了理解，将学习成果变成可互动可分享的正能量传播实践。同时，JSRC倡议并联合19家SRC共同开展“安全普法周”活动，覆盖5000多个白帽子群体，为其深度解读《网络安全法》，引导并凝聚社会共识，共建社会同心圆的良好局面，进而提升维护网络空间安全能力。

构建京东安全品牌矩阵，提升整体传播势能

◆ **外部活动。**京东安全峰会——京东安全峰会是安全界技术分享的盛会，已成功举办4届。大会集合业内顶尖的安全专家及人才，汇报京东安全建设成果，为大家提供交流分享的平台，旨在集合力量构建良好的安全圈生态系统。

安全乌托邦沙龙——JSRC沙龙已举办11届，走过全国7个城市，深受业内好评及白帽子的喜爱，如图4-31所示。沙龙成功维护深化了与白帽子及合作伙伴的关系，并拓展吸纳新的白帽子，潜移默化提升了京东安全的业内影响力。



图4-31 安全乌托邦沙龙

彰显技术能力——JSRC积极投身技术研讨分享会，紧随行业趋势，输出技术能力，合作共赢，促进整个行业良性发展。

输出技术产品——2017年，京东安全与京东云紧密合作，将京东安全团队的技术实力



和上千名顶尖白帽子注入产品及服务，保证了京东云用户安全。

◆ **内部活动**。京东集团信息安全月——每年7月，京东安全部会联合内控、合规、法务、党委及相关业务部门，共同开展信息安全月活动。为全员进行信息安全培训，提升京东全体员工的安全意识，增强大家对待数据安全的敏感性，最大程度避免信息泄露的威胁、干扰及破坏。

专家进京东活动——JSRC定期邀请信息安全领域专家走进京东，为京东人带来前沿技术分享，知识由点及面进行传播，为大家的日常工作提供了支持，提升了工作效率。

2.破冰前行——JSRC的转型与壮大

京东建立SRC团队，是保证企业对安全漏洞的即时响应，也体现了企业对用户负责，对安全重视的态度。这种态度是值得鼓励和坚持的，SRC运营投入也是十足和长久的。把现有业务做大做精的基础上，未来，JSRC会朝着SRC云化、人才培养、安全能力品牌化方向大步迈进。

◆ **SRC云化**。整合SRC资源，凝心聚力，聚焦项目，为京东云的客户专业快捷的安全服务及安全解决方案，帮助云客户避免安全问题被恶意利用的风险，为白帽子搭建更广阔的平台，促进整个互联网安全生态健康发展。

◆ **人才培养**。帮助各高校建立实训实验室、培训基地、创业平台等，实现校企间的优势互补、资源共享、人才输出、持续发展；为应届生提供攻防实战及企业实习机会，形成院校到工作的无缝衔接；依托京东安全的专业技术能力和多年安全经验，制定学习课程，对安全人才进行定向培养，实现安全人才储备，丰富行业人才资源池。

◆ **安全能力品牌化**。通过对京东安全品牌资源的梳理和规范，清晰地树立京东安全品牌形象。由点及面，总揽全局，让安全因素作为用户选择京东的重要因素，因京东安全而提升用户对京东全部产品的信心和信赖。



花落知多少：信息安全部安全专家，负责安全攻防、检测。
Lupin：信息安全部安全专家，负责安全攻防、检测。

4.6 揭秘京东大促数据屏墙

4.6.1 系统概述

1.系统定位

现如今，专业的大屏幕系统已经成为企业信息化时代的标配，它通常是指由一个个小的LED模块面板组成的尺寸超大的显示屏，假如你有幸走入京东的作战指挥中心，会看到三面高3m，最宽达18m的LED屏墙为大家呈现的数据盛宴，如图4-32所示。这场盛宴分为两个主要菜系，一个是面向媒体和公众的公关屏，一个是面向京东集团内部的综合屏。

面向媒体和公众的公关屏，视觉上要达到科技、震撼的效果，内容上要做到丰富，层

次分明，数据要做到实时、稳定、精准。正如有人说“买买买”充当着经济的晴雨表，这面屏是展现公司全景动态的重要窗口，也是展现中国电商消费行为态势的一面镜子。



图4-32 京东的作战指挥中心大屏

面向京东集团内部的综合屏，是618期间的作战驾驶舱，目的是让备战工作有的放矢，让数据成为生产力，对内容的覆盖面要求要全，从前端用户下单带来的销售和流量分析，到后端的订单生产、仓储配送以及售后的各相关系统监控，从后台对业务系统和基础运维的核心监控，到对系统安全威胁的实时扫描跟踪，全部包含在内。

2.开发背景

作战指挥中心的建立之初是为了提高研发团队保障系统稳定性、安全性和处理突发事件的能力，最大限度地预防和减少突发事件及其造成的损害，保障消费者在618期间购物环境的流畅性和稳定性，实现系统保障从被动解决型向主动保障型转变，那么屏墙无疑成为作战指挥中心信息聚合的最佳载体，凭借其面积大，显示效果佳的优点，可以同时呈现更多的信息量。但是随着研发系统信息化程度的高速发展，研发团队对可视化的需求越来越大，内容覆盖面越来越广，数据屏墙可视化产品应运而生了。

4.6.2 系统实现

1.系统架构

数据屏墙整体架构如图4-33所示，主要包括计算层和应用层。数据源包括线上业务的方方面面，如数据库日志、生产系统日志、消息中间件、自定义接口等。

计算层要求数据准确、计算迅速、系统稳定，应用层强调展现美观、内容丰富、开发方便。数据屏墙必须实现计算和应用的完美统一。

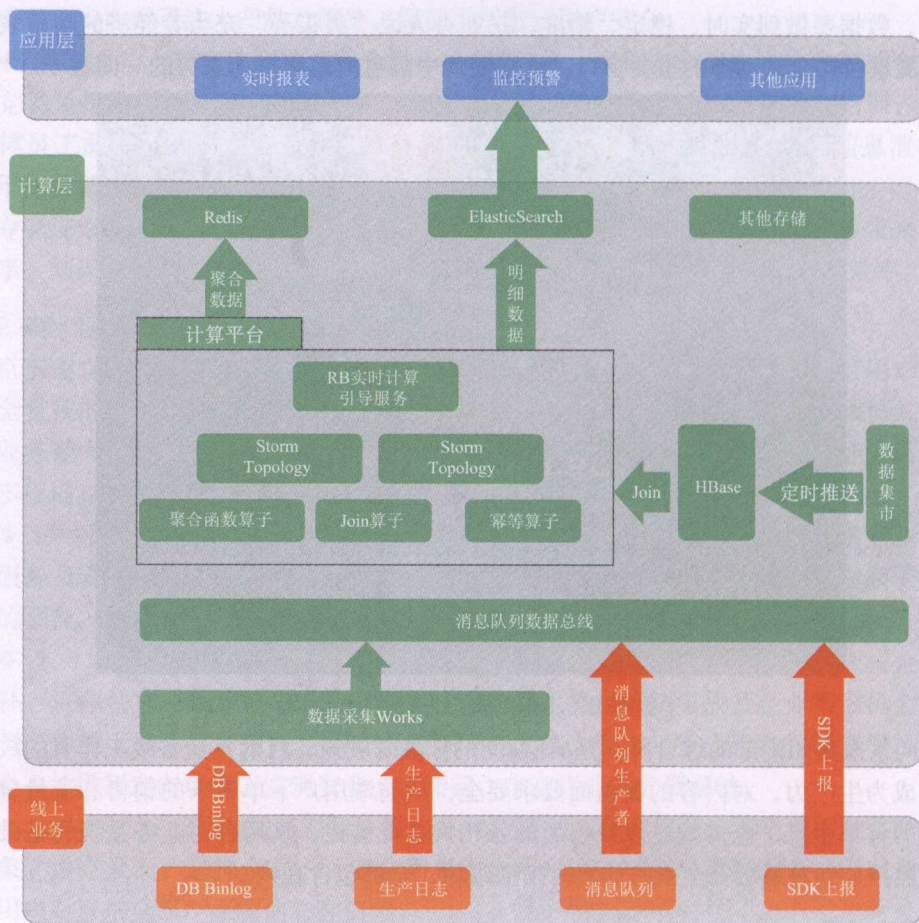


图4-33 数据屏墙整体架构

2. 计算层实现

计算层模块设计时会考虑HA方案，但难免会有疏漏的地方，例如如果依赖的某一个服务接口不可用了，或是消息队列阻塞，都会影响系统的稳定性，为保障数据屏墙真正实现高可用，团队的架构师李海龙采取了机房异地多活的设计方式，整体机房配置方案如图4-34所示。

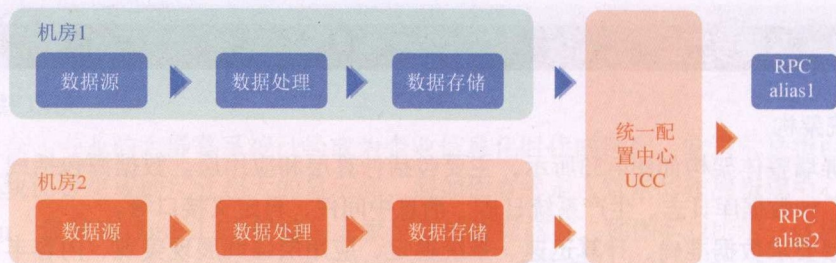


图4-34 整体机房配置方案

统一配置中心UCC是中间件部门提供的配置服务平台，可动态将更新后的配置实时推送到数据屏墙读服务的客户端，通过UCC，可以动态切换数据屏墙读服务依赖的数据源，同时数据屏墙读服务也可以通过别名来将不同等级的用户分组，在数据屏墙使用用户无感知的情况下，屏蔽系统故障。

618期间，业务部门会对数据屏墙提出很多新的需求，尤其是在临近618时，随时可能修改数据逻辑，添加数据维度，我们的系统架构如何应对，是一个极大的挑战。作为系统架构师的李海龙提出了实时计算引导服务RB（Realtime Bootstrapper）的解决方案。

RB抽象实时计算：流计算应用场景无非是实现某些聚合函数，在计算过程中会涉及动态数据关联（多流Join）、静态数据关联（维表Join）、消息幂等处理，RB将所有这些操作，包括聚合函数都抽象成对应的算法，在以后的开发过程中只要通过XML去描述业务逻辑便可以快速开发出一套拓扑结构，发布到数据屏墙实时计算平台上，基于RB将整个开发过程可简化为如图4-35所示的数据逻辑。



图4-35 数据逻辑

最终针对系统的优化，变成了对所有算法的优化，同时考虑到了合并算法以缩短中间过程。在冷热数据处理方面，架构师团队设计了内部热数据的内存处理方式，极大提高了热数据处理效率。

3.应用层实现

应用层强调展现美观、内容丰富、开发方便，落地到实现环节涉及两个主要子系统：内容控制系统和内容制作平台。

内容控制系统使系统管理员可以做到实时控制屏幕内容、快速切换应用场景，动态集成监控系统。

内容制作平台则可以方便实现图表组件插入，灵活配置图表样式，减少内容制作成本，降低内容制作的门槛。目前支持多种数据源，包括了主流的 MySQL、CSV、JSON、API等。能够提供39种组件：标题、背景、边框、装饰、iframe、单张图片、日期、视频、数字翻牌器、图片翻牌器、折线、柱状图、饼图、象形图、中国地图等。针对不同的组件，可以分别设置该组件对应数据的刷新频率，组件采用WebSocket与Ajax技术与后端通信，提供秒级刷新的实时数据可视化效果。并且可以通过拖曳的方式快速进行排版，同时预览可视化效果。

4.生产可视化实现

生产可视化是京东数据屏墙2016年开始配置的一个实时监控子系统，拓展了作战指挥中心监控范围，一期纳入了上海亚洲一号、广州亚洲一号、固安分拣中心3个大模块，二期增加了生鲜溯源和质检展示2个模块，到2017年三期规划，又涵盖了GIS分拣及配送展示模块。

系统以公司CDN网络架构为基础，视频解码技术为手段，实现了全国库房的实时画面监控，为公司各种参观、访问、接待提供了非常直观的宣传渠道，难怪数据屏墙的首席产品经理王宇瑄对生产可视化一直褒奖有嘉，每次接待非常吸引眼球。



4.6.3 系统价值

一个良好的大数据应用效果的体现，是从推开门那一刹那开始的。随着京东作战指挥中心的对外参观开放，越来越多的媒体、政府、企业、品牌商希望通过可视化手段提升数据使用体验。通过对数据进行不同形式和纬度的综合展示，360度全景展示公司业务发展的实际情况，真正做到信息零距离，不仅有效帮助集团各业务模块管理者从业务管理、事前预警、事中指挥调度、事后分析研判等多个方面提升智能化决策能力，真正帮助决策者运筹帷幄，决策千里，而且屏墙可视化将继续发挥助力作用，推动大数据在各个行业的应用。数据屏墙，让一切尽在掌握，这就是屏墙数据可视化产品的魅力和价值所在。



老虎不在家：京东商城京东视界负责人。
沉淀的果汁：京东视界产品负责人。

第5章

智能履约

瞬息万里，使命必达

千里江陵一日还。下单之后，最让人关心的是等待拿到心仪商品的过程。京东在供应链、物流方面有独特的竞争优势，让过去以天计算的配送、物流效率，变成了以小时来度量。未来，伴随着无人仓、无人机、无人车的成熟，京东配送效率的再次升级指日可待。



5.1 智慧供应链

5.1.1 智慧库存

从某种意义上讲供应链是一个库存链，商品从京东供应商或者品牌商，到京东自营仓，最后到客户手里，发现是物在流动，对我们来讲是一个库存链。如何管好库存是京东的命脉。每逢618，重中之重就是要把库存计划好、准备好，但是要备什么货、备多少货，各地备多少，这是组织者首先要考虑的问题。这时就离不开销量预测了，下面先谈供应链的销量预测。

1. 销量预测

销量预测作为智慧库存的重要一环，负责提供每日千万级别SKU到仓维度的销量预测结果。预测结果的准确度直接影响下游系统的效果和计划的成败。同传统供应链相比，京东的销量预测面临诸多难点。一是数据量大，京东销量预测系统每日需提供千万级别的预测结果，需处理的数据量达到TB级别，并且随着业务的发展还在持续增长中。二是影响销量预测的因素众多，如日常数百种促销活动、外部引流、618等，使得有些SKU的销量产生巨大波动。三是SKU间的销量差距巨大，不同的品类间的规律也千差万别，如季节品、爆品、滞销品等。四是影响销量的因素变化成本小，随时可以被人为更改，如价格、促销位置等。更为复杂的是智能推荐算法在电商的应用，每个浏览者看到都是个性化的内容，使得这些因素对于销量的影响更加难以量化。

鉴于以上问题，在实际应用场景中，任何一种单一的算法都难以满足所有SKU的预测需求。为了解决上述难题，京东的销量预测系统在设计上采取了如下几项措施：

- ◆ 销量预测系统整体采用HDFS+Spark+Python的架构，最大限度地提高系统的并行度，使得系统有能力处理大规模的数据，并按日提供所有SKU到仓维度的预测销量。

- ◆ 提取并细化各种影响销量的特征，如促销、流量、历史销量等，并且根据业务发展持续更新和增加相关特征，以便持续提高预测的准确度。

- ◆ 开发数十种预测算法并行预测，并最终由集成算法自动为每个SKU选择最佳的预测结果，提高预测结果的鲁棒性。

- ◆ 预测系统框架化，使得新的算法能够得到快速验证并快速应用到下游系统中，提高系统的迭代能力。

不管多么复杂的预测系统，一般都会分为以下几个步骤：历史数据获取、数据清洗、模型选择和训练、模型应用、模型效果监控和更新等。下面我们将按上述几个步骤详细介绍京东的预测系统。

为了让读者对京东的销量预测系统有整体的认识，本节将介绍系统的整体流程。如上所述，预测所使用的原始数据都储存在HDFS上，并用Spark对数据进行并行处理和清洗，清洗后的结果同样储存于HDFS中。算法和模型训练同样是基于Spark，系统会根据不同的算法选择相应级别的并行度，进行模型训练和销量预测。销量预测系统的整体逻辑如图5-1

所示。

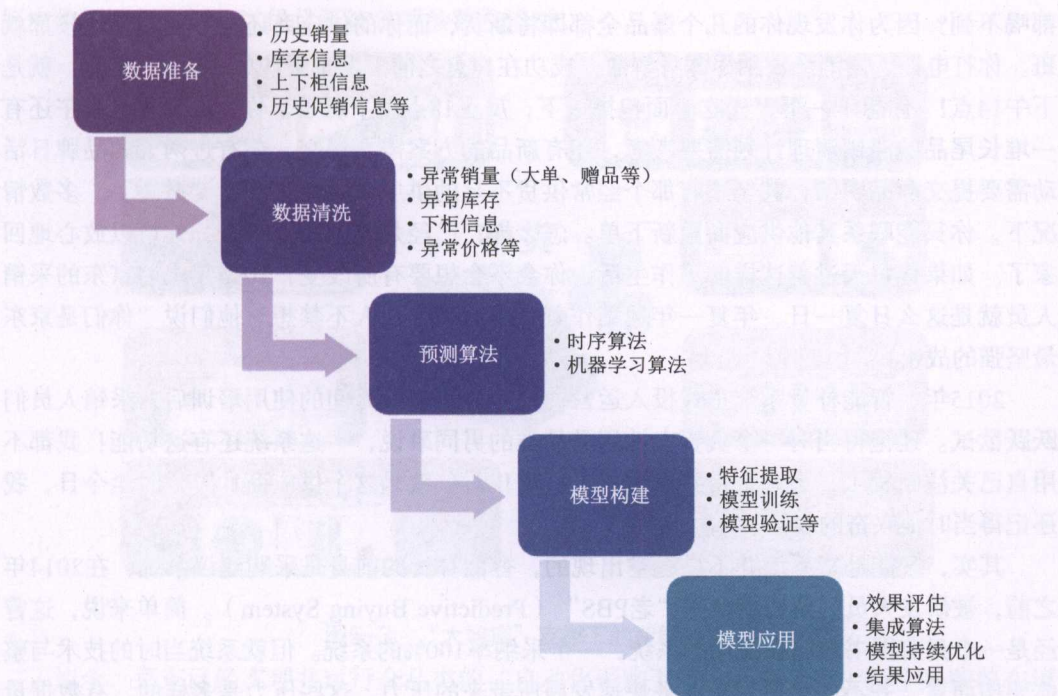


图5-1 销量预测系统的整体逻辑

在准备好相应的历史数据和特征后，接下来是用选取的算法进行模型构建。由于数据量巨大且线上系统对时效性有一定的要求，因此销量预测系统采用了大数据相关技术，开发了基于Spark的并行预测平台。所有过程包括数据获取、数据预处理、特征提取、模型构建等都被整合到这一平台被并行高效处理。特别地，团队针对不同的算法设计了不同的训练策略以最大限度地提高效率。如GBDT算法，由于是串行算法并行度不高且在训练过程中会消耗大量内存，因而对训练数据的规模有一定的限制。但是如果训练的数据过少，训练出的模型就不具有一般性，从而影响应用的效果。为了解决这一矛盾，团队设计并验证了很多措施，在实际应用中取得了很好的效果，如按照不同的品类和DC对数据进行切分，并对每份数据进行并行建模。在模型构建之前，先筛选掉一批滞销或者销量较稳定的SKU。这部分SKU的预测可由较为简单的时间序列算法承担，不会对预测准确度产生很大的影响。

在构建完模型后，接下来的任务就是如何在实际业务过程中充分利用这些模型。在Y事业部，算法模型不仅仅用来预测销量，而且已经被扩展到了更多业务场景，如单量预测、产能预测、财务预测等。举例来说，在2017年的618期间，产能预测提供全国接近7000个站点的单量预测结果，准确率符合业务方预期的站点占总站点的80%以上，为产能计划提供了强有力的支撑。

2.智能补货

试想一个这样的场景，你早上7点起床，8点出门坐上开往公司的班车，9点之前成功到



达公司打卡。伴随着一个三丁包子开启了一天的工作，为什么这么着急？为什么连杯咖啡都喝不到？因为你发现你的几个爆品全都即将断货，而你的供应商还在不明就里地按部就班。你打电话、发消息，用尽浑身解数，成功在14点之前下发了采购单，对！没错！就是下午14点！你想了一下，先吃个面包撑一下，反正18点就下班了。你太天真了，下午还有一堆长尾品的滞销清理计划需要考虑，还有新品的方案需要提交，还有过两天的品牌日活动需要提交商品明细，甚至还有那个经常供货不足的供应商小妹来跟你“撒娇”。多数情况下，你只能联系其他供应商重新下单。怎么样？已经是晚上20点半了，你可以放心地回家了。如果你每天过着这样的工作生活，你会不会想要有所改变？2015年之前京东的采购人员就是这么日复一日、年复一年地工作着、生活着，让人不禁想对他们说“你们是京东最坚强的战士！”

2015年，智能补货系统正式投入运营，在经过了一段时间的使用培训后，采购人员们跃跃欲试。还记得当时一个负责女性用品品类的男同事说，“这系统还有这功能！我都不用自己关注断货了，系统都帮我出建议了！哎！对！就是这个供应商！”，时至今日，我还记得当时他兴奋的表情。

其实，智能补货系统并不是凭空出现的，智能补货的前身是采购建议系统，在2014年之前，被研发人员们亲切地唤作“老PBS”（Predictive Buying System）。简单来说，这曾经是一个只有图书部门在使用的系统，一个采纳率100%的系统。但就系统当时的技术与整体架构而言，并不能支撑京东业务快速发展所带来的压力。这些压力是多样的，有数据量的几何式增长压力，有业务扩张带来的商品复杂度压力，最严峻的就是计算时间带来的压力，我们需要在短时间内给出最优的解决方案，这个方案就是“老PBS”华丽的变身——智能补货。

在系统重构前期，团队不知道每个功能对业务的意义，当时采购平台的胡浩指明了方向：“系统是给采购做的，采购平时怎么工作，你们谁知道？”于是，研发人员纷纷投入到一线调研，陪同、观察每一个采购的日常工作，与业务部门定期展开讨论会，做了大量的准备工作，迅速敲定了具体方案与长期规划，确定了如图5-2所示的什么是智能补货和5-3所示的“大数据”支撑下的智能补货。

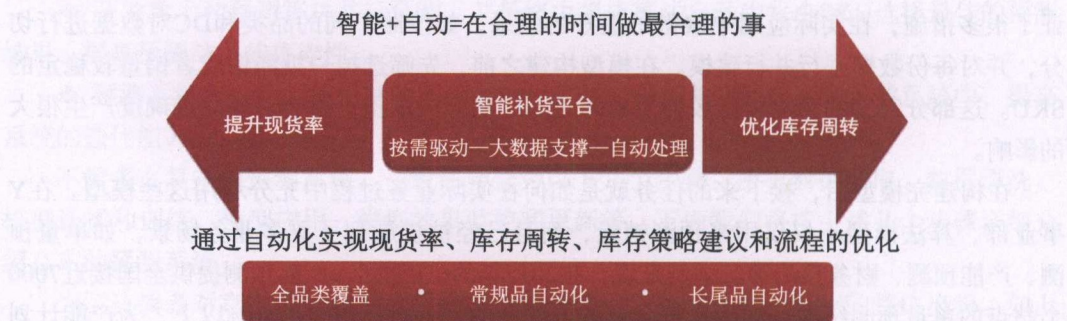


图5-2 什么是智能补货

面对每天千万级的数据计算，团队讨论后一致决定，MySQL只保存两天的数据，将数据同步到集市，这样既能缓解系统压力，还能为后续的数据分析提供帮助。智能补货系统

本身是一个计划系统，采购建议也是当天使用，作为一个数据驱动的系统，需要对大量的历史数据进行分析，以优化系统各项参数及模型。



图5-3 “大数据”支撑下的智能补货

如今，京东已经实现并运行全品类的“自动化采购下单”，每天有上万个采购单在清晨下传，采销、供应商、仓储人员在上班第一时间就可以看到并处理每一份采购单，甚至在上班的路上就可以跟供应商沟通好送货事宜。回到文章最初，假如你还是那名采销人员，早上8:40在公司大厅惬意地喝着美式咖啡，吃着经典苹果派，看着今天的采购单概览，那个总是缺货的供应商“小妹”再也没有烦过你。因为“智能补货”已经根据历史数据对比分析了每个供应商的供货情况与履约程度，准确判断出合适的供应商与合适的采购量，美好的一天开始了。

智能补货的设计初衷是为了缓解采销人员的压力，像文章开头的故事那样，系统做的是下午14点之前的工作。未来，系统会更加实时，实现根据采销人员习惯自动下单。系统会更加便捷，减少复杂的参数设置环节，上下游系统实现全面打通，一次设置，全面生效。采销的习惯、采销的设置，系统全都帮你记住，智慧供应链，聪明的不是一点点。

3. 智能调拨

为了满足日益增长的销量，进一步扩大销售区域，深入中国二、三、四线城市，京东做出了渠道下沉规划，能让更多的网购者享受到京东独一无二的“极速达”优质服务。智能调拨系统根据商品的畅销等级、商品销售预测，公司经营战略等因素，并结合京东仓储布局情况，将畅销品从区域大仓调拨到前置仓，客户下订单之后优先从距离客户最近的仓库发货，通过京东优质的物流运输以最快的速度送达客户手中。京东独一无二的建仓模式构成了复杂、高效的多层级物流网络，这种模式能够多频次、小批量的持续调拨，使库房保持极高的现货率，为客户提供“极速达”服务奠定了可靠的后勤保障。智能调拨网络布局如图5-4所示。



智能调拨是一个算法模型与业务规则相结合的计算系统，目标是计算出所有商品从哪个仓配出，入到哪个仓并且调拨量多少的问题，正如团队负责人所说：

“我们写的每一行代码将会决定京东几百万商品的走向。”不错，核心就在这里，看似一条简单的调拨建议，其中包含的关键信息很多。例如，商品编码、配出配送中心、配出库房、配入配

送中心、配入库房、计划调拨量、实际调拨量、现货库存等。这里每一个关键属性都是从海量数据里面抽取出来的、调拨白名单是从几百万的商品信息中计算出来的、库房属性基础数据是从上亿数据中提取出来的，实时库存是通过JSF中间件服务每分钟上万次的调用获取的，包括京东的大数据平台、Redis缓存、UMP统一监控平台，这些技术与服务相结合共同完成了全国RDC—FDC的配货。

618是考验系统稳定性、性能、调拨效率、调拨准确度的大好时机，也是每一位研发工程师最难熬的一段日子，那团队是如何保证618系统高效运转的呢？方法包括降低与第三方平台交互的耦合度，引入促销算法模型提高调拨精度，海量数据的存储与处理，系统监控不能少。

总而言之，系统稳定、调拨高效、调拨精准是我们作为技术人的目标，为顾客订单时效性保驾护航是智能调拨的价值所在。

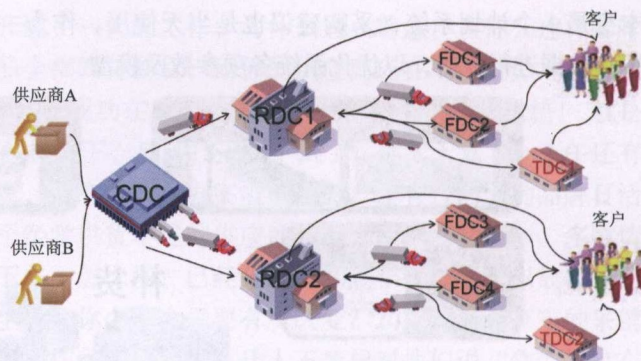


图5-4 智能调拨网络布局

5.1.2 深度协同

做好618备货，离不开与供应商的深度协同，接下来介绍供应商深度协同系统及备战工作。

1. 深度协同系统简介

供应链协同系统（Supply Chain Collaboration, SCC）起源于20世纪80年代，美国的日用品制造商突然接到一个超市发来的需求，它说每次补充新的尿布都需要经过一次订货手续，非常麻烦，可不可以当尿布没了，新货就补上？制造商当时很重视这个需求，把两家计算机连起来后，做了一个自动补充尿布的系統，结果试用起来效果良好。这套系统逐渐得到了扩充，加入了很多商品种类，最终形成了供应链协同管理系统。

21世纪是信息化时代，市场竞争日趋激烈，供应链协同不仅体现了信息化的特点，也帮助企业之间实现了组织层面的协同、业务流程层面的协同以及信息层面的协同，从而共同面对市场竞争，积极快速地响应市场需求，实现合作共赢。

早在2007年，京东就开始自建物流体系，以打造从购物下单，到仓储配送一体化的体

系，其中供应链深度协同系统更是发挥着不可替代的作用。随着亚洲一号的投入运行，协同系统越来越体现出它的效能，它提升了与供应商的协作，加速了商品流通，可以第一时间把客户订单分配到对应仓储，从而实现第一时间送到客户手中。京东仓库补仓流程如图5-5所示。

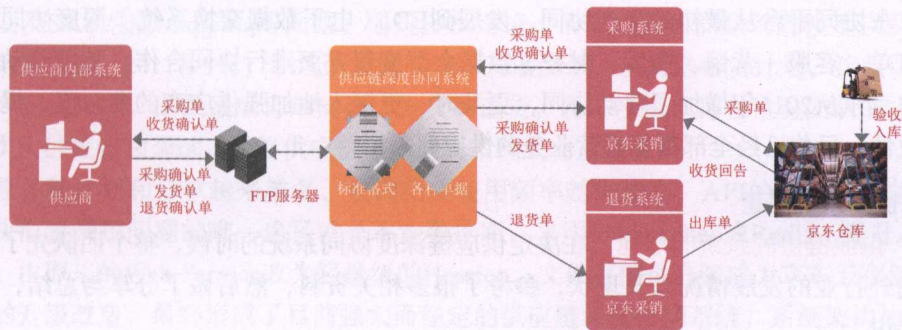


图5-5 京东仓库补仓流程

2. 京东协同平台简介

协同平台是京东面向自营供应商的门户，平台统一入口、统一权限配置的分布式网站系统。面向京东自营供应商提供灵活、快捷、流程化的一站式管理服务。并对接京东商品、采销、售后、财务等多个业务体系，实现了规范化的业务审核流程、各业务维度数据的实时查询以及针对各业务维度的待办实时提醒、预测与分析。真正成为京东整体供应链中与供应商衔接的主要环节。协同系统包括简单的业务协同和双方系统交互的深度协同两部分。

协同平台自2011年创建以来，从开始的以协助采销人员服务为主的工具化版本，先后经历了三次版本迭代，逐渐面向以提高京东供应链协同深度为主，打通供应商与京东之间的桥梁，指导和监督供应商的服务，规范化业务以及系统可靠性，为后续各业务线发展奠定基础。协同系统发展到现在，整体业务如图5-6所示。

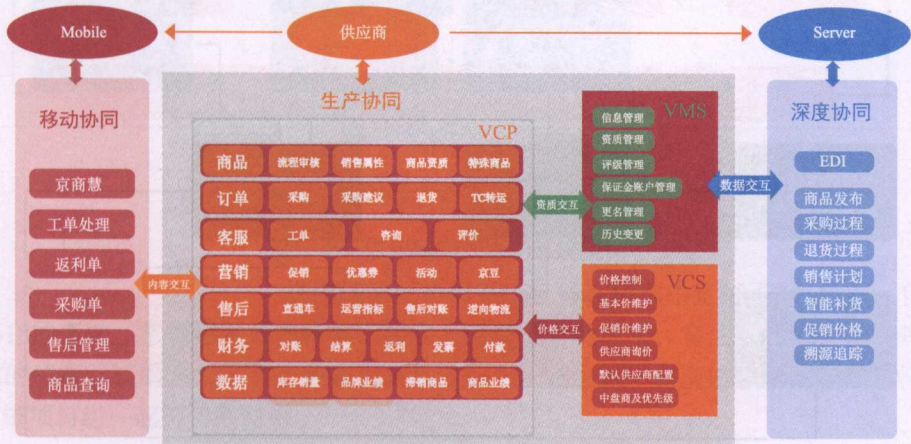


图5-6 系统范围拓扑图

京东一直在加强与供应商的合作，在业务合作深度和广度方面都有很好的表现。目前



京东正式合作的供应商有近万家，这些供应商分别与京东现有的“家电”“3C”“消费品”“服饰家居”“生鲜”“图书”“新通路”事业部进行着日常业务往来。供应商作为京东自营业务的合作伙伴，与京东进行着深度的业务协同工作。同时，京东本着开放、高效的原则，为供应商打造了完善的业务协同系统。

京东协同平台从最初的生产协同，发展到EDI（电子数据交换系统）深度协同，从商品、订单、客服、营销、售后、财务、数据全部流程方面进行协同合作。随着移动业务的普及，京东从2016年增加了移动协同，更及时、更深入地加强供应商的参与度，提高协同的高效性，目前已经全部覆盖自营业务的供应商。

3. 协同系统搭建

◆ **供应链协同系统的搭建。**在决定供应链深度协同系统的时候，整个团队先了解了目前国内外行业的发展情况以及现状，参考了很多相关资料，然后做了分享与总结，达成了一定共识。

首先解决的是协同系统的权限目录，以及当时的采购业务和数据业务，针对如何控制权限，保障数据安全性，反复设计了几种方案。最终确定以复用京东账号登录，自己控制权限的方式，从安全、业务各方面得到支持。2011年4月，协同系统问世，标志着京东协同系统的诞生。

◆ **供应链深度协同系统的搭建。**2011年7月，由于图书商品的特殊性，恰逢图书频道上线不久，当时的图书供应商覆盖了几乎所有的中文图书品种，有100多万种的图书产品待接入。团队决定启动供应链深度协同系统的开发建设。当时，每年还有很多供应商加入，而且每年图书出版业有20万种左右的新品种出版，图书产品线新品种会快速迭代添加。因此为了保证所有海量的品种都可以快速上架，团队决定页面信息优先做到完善和准确，这样才能带来大量的流量和搜索点击率，以实现用户快速定位自己需求的商品。为了达到这个目标，团队采取了自动化商品信息处理的方式来发布商品信息，整体系统处理流程如图5-7所示。

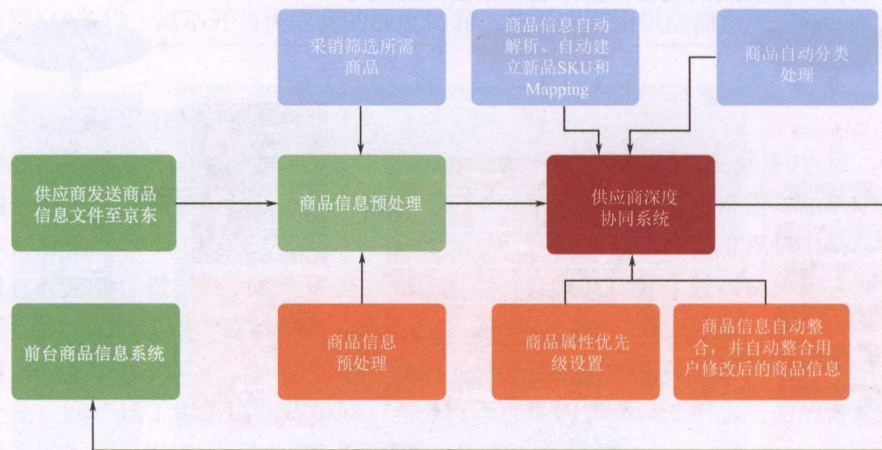


图5-7 整体系统处理流程

面对千万级别的数据量，我们首先选择了分库分表以及分布式MySQL数据库。在应对

大量元数据文件读取和解析时，采用了按照文件大小平均分配给分布式任务调度框架去同时读取和解析。

确定方案后，项目负责人带领项目组进入到了紧张的开发阶段。由于项目工期比较紧张，项目组直接定一个会议室开始一期需求的封闭开发，处理商品信息自动解析、自动创建SKU以及SKU对应Mapping等问题，通过多次讨论，形成成熟的商品信息模型以及SKU生成策略等方案。2011年9月，系统研发完成进行灰度测试，从发送备货计划，接收供货计划，排产到预采购，以及最后订单确认、发货和退货，每一步操作都在检验团队的成果。经过一次次的报错和调试，最终供应链深度协同系统成功上线了。

随着各种计划和订单越来越多，查询功能使用频率越来越高，API的调用率也急剧上升，解决海量调用问题的唯一途径就是不走数据库，采用Memcached或Redis，提升API调用响应。由原来的Web Service改为轻量级的Hession，文件存储也升级成为京东云存储。经过多次的升级改造，最终形成了目前强大而稳定的供应链深度协同系统，系统架构如图5-8所示。

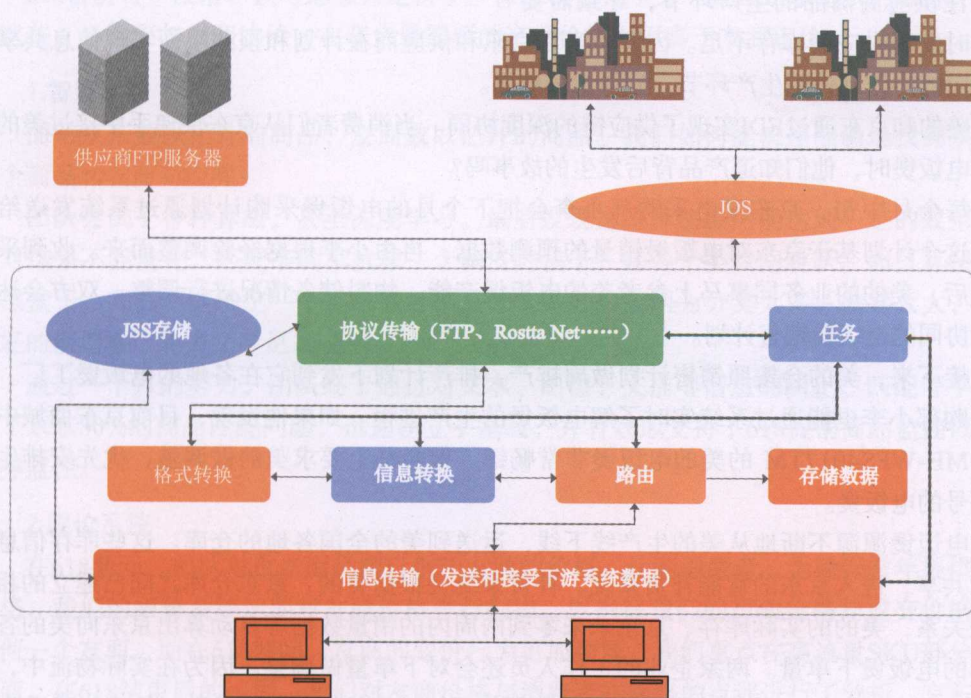


图5-8 供应链尝试协同系统架构

◆ **路在脚下，展望未来。**供应链深度协同系统已经在风雨中走过了7个年头，它经历了7个618的考验。从懵懂到成熟稳健，很多人见证了它的成长，有的人在深夜里抓耳挠腮解决它的病痛，有的人大年三十还在为它添砖加瓦。它凝聚了供应链项目的辛勤和汗水，是供应链项目组一起努力的结果。

展望未来，供应链深度协同系统还有更长的路要走，在平台开放的大潮下，深度协同系统未来也将开放给设计，为社会大众提供服务、带来便利，提升企业的生产效率。供



应链深度协同系统将完成开放服务平台的改造建设，提升平台自助服务能力，开放并扩充现有协议，对外提供系统接入和预测、计划等服务，如图5-9所示。

4.深度协同——电饭煲之旅

2014年年底，京东和美的达成了战略合作关系，双方在物流配送、大数据分析、智能设备等方面进行深度合作，其中实现供应链深度协同也是战略合作的重要内容之一。

在对京东和供应商协同的实际业务调查时团队发现，业务多关注采购环节，较少关注供应商商品的生产环节，导致需要补货时发现供应商库存不足。因此，实现京东和供应商在计划和预测层面实现信息共享，将供应链协作延伸到生产环节，显得十分重要。

美的和京东通过EDI实现了供应链的深度协同。当消费者们从京东快递手中接过美的品牌的电饭煲时，他们知道产品背后发生的故事吗？

每个月中旬，京东家电采购部小李会把下个月的电饭煲采购计划通过系统发送给美的。这个计划基于京东对电饭煲销量的预测数据，再由小李根据经验调整而来。收到采购计划后，美的的业务同事马上参考美的电饭煲产能、物料储备情况进行调整。双方会达成一个协同的电饭煲销售计划。

接下来，美的会按照销售计划做周排产，排产计划下发到它在各地的电饭煲工厂。京东采购部小李也能通过系统实时了解电饭煲的生产进程。如果他发现，目前京东商城中型号为MB-WFS4017TM 的美的电饭煲非常畅销。他会马上要求美的做调整，优先安排生产该型号的电饭煲。

电饭煲源源不断地从美的生产线下线，运送到美的全国各地的仓库。这些库存信息也自动共享，进入京东的智能补货系统。补货系统会根据美的、京东仓库之间已建立的相互支援关系，美的的实际库存、京东未来零到两周内的销量数据等自动算出京东向美的各个仓库的电饭煲下单量。两家企业的工作人员还会对下单量做调整。因为在实际物流中，货车有60方、80方、120方的运载能力，人们要把系统计算的结果调为整数，便于物流运输。

调整后的电饭煲补货建议自动进入京东采购系统。京东采销人员只要要在系统中单击“审核”和“确认”就可下单。接到订单后，美的同事马上在系统中进行发货预约。他们预约的送货时间要尽量避开京东各个仓库的繁忙时段。之后是送货和收货后的回告。

这样，一个电饭煲在美的和京东两家企业间，完成了一次协同之旅。整个协同过程如图5-10所示。

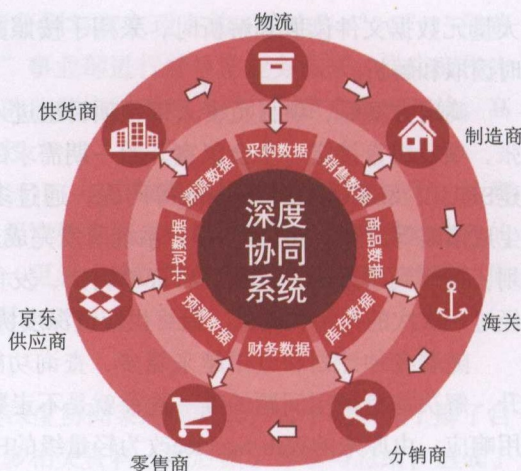


图5-9 深度协同系统发展规划图

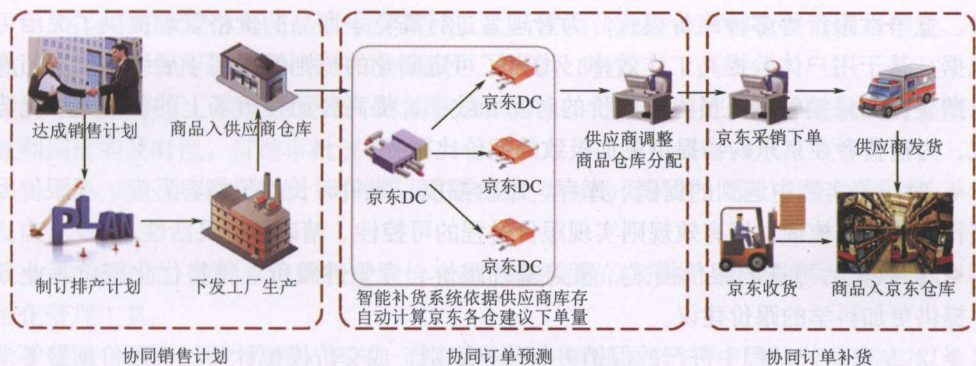


图5-10 京东与供应商的交互过程

5.1.3 智慧定价

618备货告一段落，该考虑怎么定价了。保持价格竞争力一直是京东追求的目标，要做到这些，必须要与友商比价，在客户最需要的产品价格上不高于友商定价。

1. 智慧比价

面对京东数以亿计的商品，友商数以亿计的商品。我们如何能快速准确地找到京东的某个商品和友商对应呢？

团队尝试了各种算法，甚至深度学习。最后发现这些算法虽然能达到一定的效果，但是对结果不能量化。匹配准确率就像一个调皮的孩子，在各个类目间跳跃。而且算法是一个黑盒子，我们不能确定一个类目做好了是怎么做好的。因为分类太多，差异太大，而训练好的模型推广能力又不足。

通过一个月的努力，团队终于通过对文本、图像、文描等信息的纠正、匹配等手段解决了京东80%的商品匹配问题，迅速建立了基线，并有效地支持了618促销商品监控以及优惠券监控。

2. 跟价系统

在618期间，团队对重点商品制订低于友商的价格，让利消费者。为确保常年提供最低价格，我们的跟价系统定期对友商价格进行扫描，以确保我们的页面价格与成交价格低于任何一个友商。而在618期间，友商的变价行为更加频繁，我们重点在高流量SKU部分对标友商。在618结束后的一周，我们对友商价格与消费者在网络的点评进行了分析，京东的价格在618期间依然是各大电商最便宜的。

诞生初衷。目前跟价SKU数量较多，手工逐个比对容易出现错误。跟价SKU对友商调价后的反应速度不及时，因无法及时跟踪导致销售机会流失。对于在什么情况下无须跟价缺乏明确的规则，导致决策过程冗长和反复。管理层对比价跟价的整体过程无法有效进行监督和控制，往往存在一定的主观性和滞后性。

结合人工经验的智慧系统，系统提供系列动态交互规则，将传统手工跟价过程中重要但易忽视的库存、促销、供应商管控等因素一并纳入，创新性地设计了弹性跟价、容忍度



跟价、竞争性跟价等多种跟价模式，为管理者进行高竞争商品的价格管理提供了灵活可控的依据。基于用户体验提高工作效率，增设了可定制化的规则配置用于后续迭代及拓展。项目期望提高采销管理人员人工比价的时间和效率，提高公司在市场上的价格竞争优势和形象，为消费者在京东购物提供更加极致的性价比服务。

◆ 对跟价实践中遇到的促销、库存、跟价幅度、跟价时长、价格容忍度、不跟价因素等进行业务流程梳理，用有效规则实现跟价过程的可控性、精确性、灵活性。

◆ 提供灵活可选的跟价模式，覆盖基础跟价、竞争性跟价、销售优化跟价等业务场景，提供更加科学的跟价建议。

◆ 对友商竞价过程中进行的促销进行价格识别、成交价模拟计算、页面价预警等多种管控手段，并且通过价格阈线对自动跟价价格进行风险防控，确保跟价自动的同时保障安全性。

技术特点。跟价系统的技术特点如图5-11所示。主要实现商品自动化比价跟价，定制化用户比价跟价需求，有效监测友商的商品价格变化信息，实现商品跟价的快速、精确、及时，确保价格形象及竞争力，做到不遗漏、不错失高Band商品销售机会。提高商品比价跟价管理效率，充分节约采销管理高Band商品比价跟价时间，全面提高价格管理的科学性和有效性，充分考虑PDCA循环机制，前置价格发布审批环节，可滚动验证跟价有效性，为管理者提供可以精确控制的跟价参数和跟价规则，更好地达成商品销售管理目标。全面优化商品比价场景，充分提高高Band商品在市场环境下灵活变动价格的能力，同时确保不盲目跟价、跟价价格最优化。融合价格系统多平台一站式解决方案，提升高Band商品价格自动化率，提升商品智能化价格覆盖范围，一站式解决抓价、比价、跟价、调价、销售回顾的价格管理流程。通过追求“跟价精细化”的目标来制订“安全管控手段”，再通过实施“安全管控手段”逐步做到“规则完善化”，从而进一步优化“跟价精细化”。

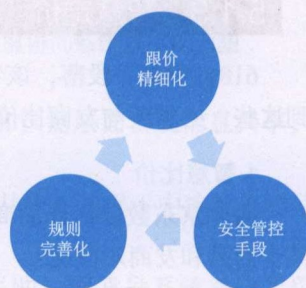


图5-11 技术特点

从图5-11也可以看出，本系统是供应链收益管理部门的“智能定价系统”与“比价系统”的一个成功结合。智能定价系统为比价系统输入了更多的十分具有比价意义的SKU进入其比价池，使得比价池更加丰富但同时维持在一个稳健的水平。而比价系统则为智能定价系统不断更新友商价格，实现智能定价引擎的改进，如图5-12所示。

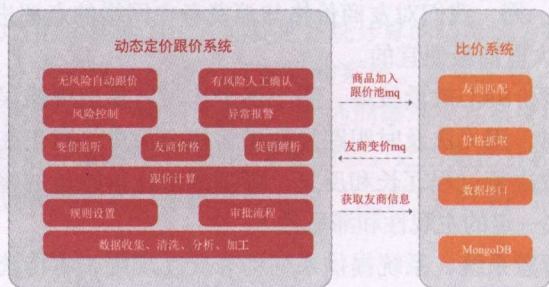


图5-12 动态定价跟价系统与比价系统结合

◆ 结合采销管理实际业务，设定规则，监听友商价格变化。解析友商价格信息、促销信息、实现成交价翻译。实时抓价的准确性、跟价及时性。

◆ 友商降价了，我们需要第一时间获取到信息，并及时跟进价格，这就需提高获取信息和调价的及时性，打通审批流程。

◆ 当友商正在进行满减、满件等与价格相关类型的促销时，我们只能根据经验手动换算友商成交价。

◆ 友商降价，我们在多大程度跟价有利于既保持价格形象，又不损失太多利润，需要精细化管理工具。

◆ 维护价格形象多数情况是自上而下发起的，管理层对于执行情况的把控多以结果为导向，对中间执行过程缺少监督工具。

实现的效益。以前手工跨平台查价、比价、跟价、调价，现在跟价一步自动化完成，轻松实现高竞争性商品的跟价目标。以前人工经验判断考虑范围有限；现在细化跟价场景，充分考虑新品、库存、促销、价格限制等多种因素，针对不同环境不同场景给出更加有针对性的跟价建议。以前比对方低即可，现在引入价格弹性、价格容忍度等新指标，将跟价程度进行更加精细化的控制，在确保价格优势的前提下，更加全面地考虑商品销售和利润置换效率。以前出现跟价过低，现在审批前置，将有可能触发价格风险的各类因素，提前进行卡控，设置了价格调价幅度、最低毛利率、价格阈限、低价审核、价格发布等多重规则，并明确了审核机制和审核流程，增加预警和通知功能，确保价格发布的安全性。

借助比价爬虫能力，监测友商信息变化，一站式抓价、比价、跟价、调价、评估，实现快速跟价、精准跟价。提供动态交互规则，纳入库存、促销供应商管控等因素，准确解析友商促销，结合商品弹性，灵活进行动态优化跟价、容忍度跟价、竞争跟价。在618期间，本系统成功帮助众多事业部提高了比价效率并且节约了人时用于其他工作。

3. 智能定价

自动定价系统解决了京东自营长尾品多、采销无暇顾及、定价不合理和长时间不调价的问题，为众多事业部提供定价支持。创新地打破周期定价，监控商品促销、价格，基于历史数据和实时数据、商品多级弹性，使用多目标优化和鲁棒优化，结合业务目标和约束，实时智能定价。

◆ **数据模型。**两年积累的历史数据，KVI模型、商品弹性、随机森林弹性、品牌品类弹性，多目标优化模型识别定价潜力，制订GMV毛利最大化价格。鲁棒优化模型具有很好的抗干扰性，在长尾品历史数据少、不确定的情况下仍能求得最优解。

◆ **实时调价。**根据商品价格、促销信息智能维护，结合变化消息智能选择定价时机。根据商品生产情况选择合适的定价模型。探测促销状态，未来有促销时也能安全定价，提高定价覆盖度。纳入促销门槛、优惠券门槛、尾数等考量，保证现有业务一致不冲突。

◆ **风控安全。**校验成本、库存、销售目标，保证价格安全可靠。黑名单商品不定价。维护历史价格，保证定价幅度。维护价格保鲜期，保证价格稳定不频繁变动。做到一键恢复。

◆ **价格决策管理。**系统自动发布无风险的价格，有风险时提醒采销手动调价。提供规则管理、商品池管理、价格监控、效果监控、价格模拟、促销监控等功能模块，采销可以



发布维护商品，查看商品信息和指标，清晰了解商品定价策略，一站式管理价格。

5.1.4 智慧履约

货已备齐，价格策略也确定了，如何使用库存呢？这涉及订单履约过程。

1. 订单履约中心

618时，漫天的订单如暴雨般灌注到交易平台，又随着交易的完成，形成巨大的洪峰向着各生产系统呼啸而至。在如此大规模的流量面前，这些系统无法独自建立有效的应对措施，纷纷宕机……面对这悲催的现实，各生产系统不禁高呼：“就不能建一道防洪大坝将流量平滑下来吗？！”

当然可以，而且现在已经建好了，它就是供应链履约中心。当然，供应链履约中心的发展并不是上面的小故事汇总所说的那样，是突然就要成立的，也是经过了一步步的演进才发展起来的。下面就来介绍供应链履约中心的发展历程。

2003年京东网站创立之后，迎来全国电商的快速发展，京东的业务随之不断增加，导致相应的业务系统也在持续增加。直到2011年，随着系统增多及业务的需要，逐渐成长出来一个小团队。他们负责在各个系统之间进行数据的传输，将客户订单数据传到库房，同时需要将非客户订单，如采购单、供应商、内配单等20多个业务传输到相应的业务系统，至此供应链履约中心成形。

在项目初期，由于各个系统的不完善，导致数据传输总是存在各种各样的问题，订单总会卡在这一环节。而作为传输环节又必须保证数据能正确传输完毕，这就需要团队必须了解上下游系统的业务及数据处理的过程，只有这样才能找出问题产生的原因，才能知道该如何去处理。但是由于上下游的系统是需要经常上线新需求的，只有每天及时了解上下游系统业务的处理，才能保证我们的环节没有问题。

随着业务复杂度的提升，系统也变得越来越臃肿。经过不断地积累经验，在对业务更熟悉的基础上，将系统分解为：订单拆分、订单转移、订单流程三个系统。而分解后的三个主要系统，也发展成为了供应链履约中心的基础。分解后的系统如下：

◆ **订单拆分**。正如字面意思，订单拆分就是将用户下的订单，根据其生产属性的不同，比如生鲜订单、大家电订单、图书订单等维度进行拆分，以便于生产。

◆ **订单转移**。订单转移则是根据拆分完的订单，结合现有的库存情况和用户的期望收货时间，来统一调配最优的生产方式。也就是什么时间去哪个库房进行怎样的生产。

◆ **订单流程**。订单流程也就是OFW包括订单下传和回传。主要是对转移之后的订单，调用库房相关的预分拣、打包等十多个服务环节，进行订单下传和生产。在生产和配送过程中，还需要将订单的相关数据进行回传，以便用户能随时追踪订单的生产情况。

基本的业务流程图如图5-13所示。

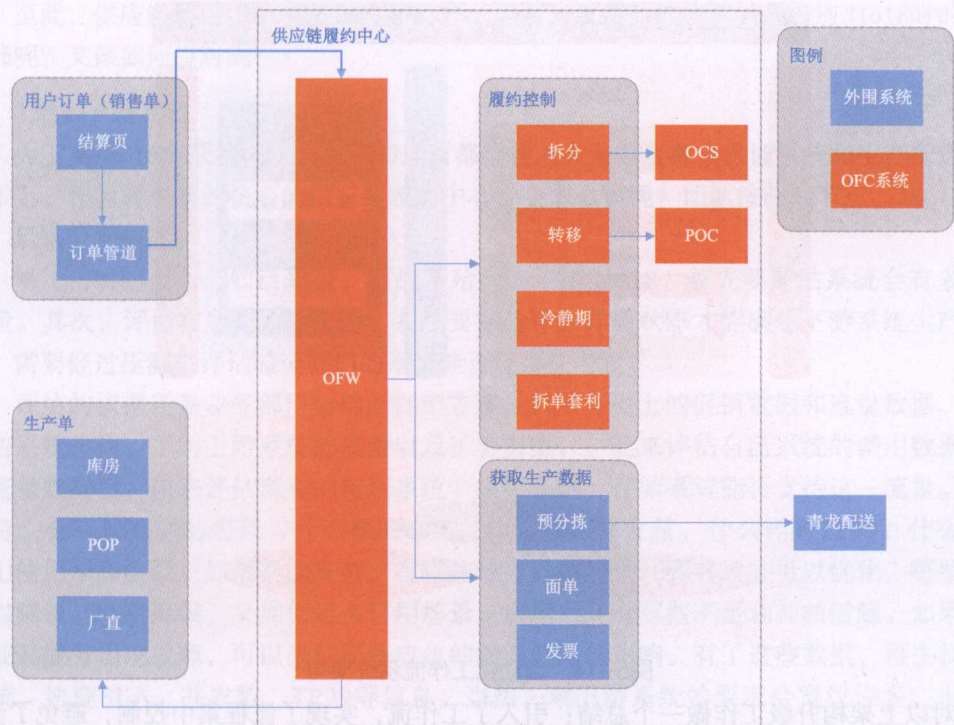


图 5-13 订单履约中心基本的业务流程

另外，在技术上，为了保证数据在传输过程（主要是下传和回传）中的准确性和及时性，保证生产系统流量的稳定性，架构师提出了要对系统重构，组建流量可控的、易运维的、分布式环境下的自动化工作流的设想。

其实，最初的OFW也不是自动化的工作流，是靠着任务驱动的工作流。也就是靠着定时执行的任务系统，通过不同的任务类型，抓取不同状态的订单，来完成整个订单生产流程的流转。但是其缺点很明显，无法提速，过于依赖DB。

改进后的OFW用了任务队列的方式，在接收订单后，识别出其需要执行的任务类型，调用相应任务队列的入队操作，然后执行相应的逻辑处理，并在处理完成后出队。订单出队后，系统会根据当前的订单任务信息判断其下一个任务是什么，然后对下一个任务队列做入队处理。并在此基础上建立了异步的任务调度体系作为补充。异步的任务调度，简单来讲就是针对在队列中处理失败的订单，按照抓取时间和重试次数定期抓取执行失败的任务，重新调用相应任务队列的入队操作来完成的任务的重试。使用任务队列之后，就能尽可能地脱离DB，全内存计算，执行效率有了大幅度提升。

基本的系统架构图如图5-14所示。

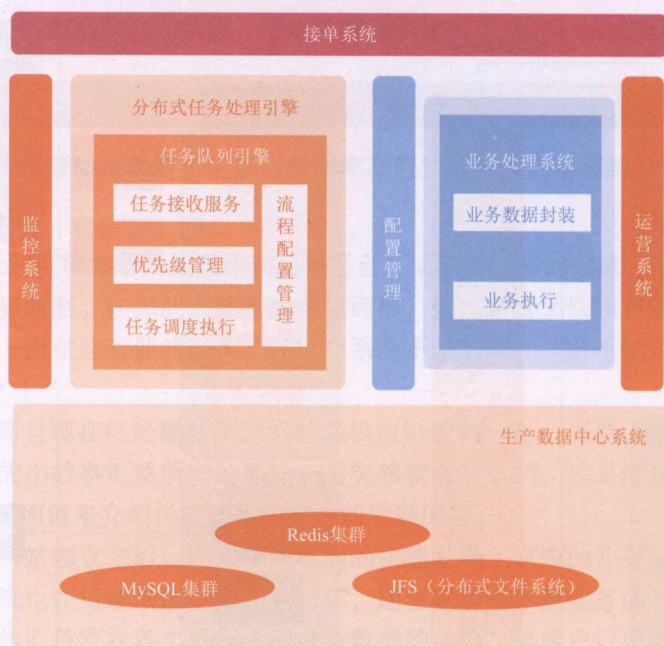


图5-14 自动化 workflow 系统架构

对以上架构升级工作做一个总结：引入了工作流，实现了流程集中控制，避免了流程分散造成的数据不一致和流程混乱现象。与“流程集中控制”相对的是“流程分散”，即流程由各业务系统本身控制，由于业务系统关注于业务本身，往往难以兼顾流程逻辑。

客户端流量控制技术。该技术也是在引入工作流过程中实现的。在工作流的前端是接单系统，它的职责是接收从上游传来的订单数据，也就意味着会受到618订单洪峰的冲击。从历史上看，这个洪峰最高达到10000次/s调用。而供应链履约中心通过客户端流量控制技术，把洪峰削平了，下游十几个系统只需要按平滑后的流量来建设系统，大大节约了系统建设成本。通过客户端流量控制技术，供应链履约中心还可以在运行时维护整个订单履约流程流量的稳定性，保证各个生产终端（例如各个仓库）可以按约定的节奏均衡地生产。

分环节内存队列技术。通过使用“内存队列”技术，以及流程中各系统服务的SLA达成，将订单履约工作流时效从原来的30min缩短到了秒级，为保证定时达（1天内可以选择三个配送时段）、极速达（2h内将商品送达）等业务提供了保障。前不久，京东创造了从下单到送到用户手中最快10min的世界纪录。“环节”指的是工作流的环节，每个环节都有自己的任务队列，这样做不是为了速度，而是为了控制，包括流量控制、流程拦截、优先级调整等。

监控与报警方案。除了接入京东的统一监控平台实现基本的监控功能外，还根据业务特点，实现了任务积压及趋势、有积压情况下处理能力的趋势、异常数据等方面的监控，报警前加了一定的分析功能。例如，如果系统出现了积压，同时系统处理能力在不断上升，说明这时出现了业务高峰，而不是系统出问题了；如果系统吞吐量在下降，响应时间正常，且没有积压，则说明是业务量在降低。

至此，供应链履约中心的基础已经打好。但是，改造后的体系，能否应对618时的流量洪峰呢？又该如何应对呢？

2. 履约大促备战

为了支撑618的顺利进行，技术和运营都会投入大量的资源来保证系统和生产配送。在技术上，作为客单履约核心的供应链履约中心，会怎么做呢？团队按时间节点，从618前、中、后来分别叙述。

◆ **618前的工作。**知己知彼，百战不殆。为了应对618，首先要评估系统会有多大的流量。其次，评估在这么大流量下，系统要保持什么样的状态才能满足下游系统生产。最后，需要经过压测来评估验证我们的系统能否经得住考验。

评估的依据还是业务部门给出的促销方案，以及历史上的促销数据和流量数据。并同上游系统沟通，了解上游系统的流量以及扩容方案，一起来评估自己系统的调用数据。有了流量数据后，再来评估现有的应用系统、缓存系统、存储系统能否支持这一流量。如能支持，会有什么样的表现，什么样的QPS，什么样的并发量，什么样的TP99，什么样的CPU使用率和负载。如果不能支持，有哪些地方需要改进，哪些地方可以优化，哪些地方可以降级。针对降级，又如何定义启用场景，启用后又有哪些不足和补救措施。如果某系统或某服务出现故障，可以采取哪些应急的措施能降低影响。有了这些数据，再去找下游系统，协商SLA、并发数、TP99等信息。当然，对下游系统的要求会宽泛许多，也要协同考虑仓配系统的实际生产要求，根据木桶原理定义出每一个系统的并发数和调用量的限制。

在评估完成后，需要根据评估结果进行线上的压力测试，检验现有资源情况能否满足618的需要。由于线上一直都有用户订单数据在执行，因此压测时间就选在了用户相对较少的凌晨时分。压测的方式，主要是模拟大量的调用，以较高的并发量来调用某个系统，观察该系统的表现。对于某些特定的系统，还可以采用减少线上应用实例个数的方式来辅助压测。压测之后，就能得到某个量级的冲击下各个系统的表现，评估出是否能满足本次618的需要。如果无法很好地满足，则需要依据生产能力来评估，最低调用量要满足多少，以及相应的降级措施和之后的系统改进方案。

对于定义出的降级方案和应急的策略，需要整理成翔实的、可操作的文档，并通过实际演练的方式让团队中的每个人都熟练掌握。除了面向整个团队的宣贯之外，还采用问答考试的方式，线上的Chaos Monkey演练等方式来确保每个人都能掌握。

◆ **618中的工作。**618的时候，大家肯定都在买买买。那么电商的研发在做什么呢？也在买买买吗？这个问题的答案，只有两个字“值班”。

看起来很简单两个字，个中滋味不足为外人道也。开始的时候，整个运营监控体系并没有完善搭建，只有被动报警这一种方式了解系统的运行状况。为了能查找隐患，只能采用SQL查询的方式来观察系统的处理能力。经过多年的努力，目前供应链履约中心已经具备了全流程可视化观测、软硬件异常报警、人工履约调度，完善地集观测、预警、控制于一体的分布式环境下的自动化运营监控平台。不仅能提供系统运行状况的数据，还可以提供业务运营的数据，用于多样性的数据分析。



运营监控平台系统关系图如图5-15所示。

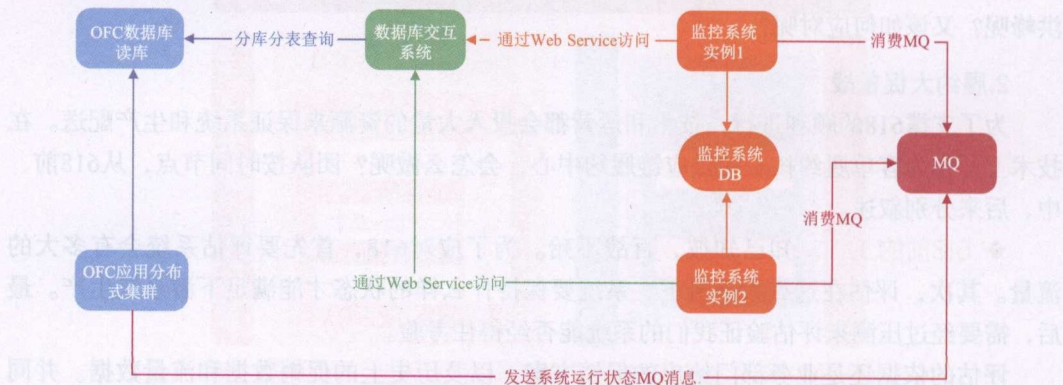


图5-15 运营监控平台系统关系图

如图5-15所示，运营监控平台的系统，主要是通过分析分布式集群发送的MQ和通过DB查询两种方式来系统运行数据进行分析。经过分析，能得到什么呢？

经过对运营数据的分析，团队发现取消订单的动作大都发生在下单后的10分钟内。基于此，供应链履约中心内部增加了智能冷静期的履约调度环节。简而言之，就是大部分订单在不影响客户订单履约的前提下，进入一个“暂停处理”的状态，等过了某段时间后再去生产。这样可以解决生产过程中客单取消浪费产能的问题。

当然，618时主要工作还是要监控系统状态，在出现异常时进行应急操作。每次618，整个商城的研发都会派出代表入驻“京东集团618作战指挥中心”，以便在出现状况时及时协调处理。这时，就需要对系统的应急预案和降级措施非常熟悉，能做到胸有成竹、信手拈来。例如，下游某个系统反馈DB压力过大，需要限流。这时就要团队快速响应，按照制订的预案，调低针对该系统的并发数，并且减少调用频率，加大休眠时间，同时观察系统整体的处理量是否有大幅度下滑，同步将相关的信息反馈给上级领导和系统负责人。不过，只做到熟悉还是不够的。在这种紧张的气氛下，个人的思维会受到很大的制约。只能依靠团队和规范才能确保万无一失。具体方法就是：值班人员按角色分工，一名总指挥分析问题给出处理方案的编号（之前梳理的应急预案和降级措施都有相应的编号）。两名执行者按照处理方案进行Double Check后执行操作。在过去的几年里，多个案例证明了这种角色分工的正确性和必要性。

对系统状态的监控，除了自己的系统，还要关注下游的生产系统，同运营同事一起根据仓配的产能来协调订单流量。为了保证仓配的生产效率，供应链履约中心目前是按生产波次来传输订单数据给仓库下发订单的。实际上，仓库的生产能力会比预计的要高。这就有可能造成某几个仓库已下发的订单流量无法完全满足仓库的生产能力，这时就要求供应链履约中心将还未下发的属于该仓库的订单及时下发下来。针对这个场景，目前供应链履约中心提供了工具来调配某库房的订单立即下传。2017年的618，运营人员就多次使用该工具调配生产节奏。从这也可能看出，坚持运营监控平台的搭建和完善，是一件长远且必要的工作。

◆ **618后的工作。**在618结束后，大家肯定都觉得应该可以休息了。其实，还是有很多

工作要做。

618时的订单量比平时提升了几十倍，对应的工单量也提升了几十倍。供应链履约中心作为交易和生产间的枢纽系统，是工单咨询最多的部门。这也能体现流程管理的益处：提升规范和效率，全面监管，积累经验并优化管理。

除了工单的咨询和协助处理，我们对618中的运营数据的分析，还发现了618中针对销量预测和生产能力预测的不足。接下来，团队将协同Y事业部的其他兄弟部门，利用我们在618中的相关数据，进一步优化预测的准确度，并同时提升针对产能和用户预期的订单智慧下发的能力，进一步提升效率和客户满意度。

3. 订单驱动补货

客户下单时被告知所购商品预计何时送达，有时这个时间不是通常的当天或者第二天送达，也许会超过2天。实际上这种情况往往是库存不在本地，或者供应商还没有把货送到京东库房，这也就有了订单来驱动补货的问题。但从哪儿往哪儿补？补多少？什么时候补？这都是订单履约系统要解决的问题。

我们首先要知道订单数据来源，因此建立了“暂停订单数据中心”，通过这里可以清楚地看到哪些订单还在暂停，原因是什么，而我们只要把缺货的订单拿到，就可以作为基础数据。

有了订单基础数据，还需要知道什么？答案是支援关系网络，它提供了往哪儿补的基础。目前的支援网络包含了自营仓、商家仓的全网覆盖，分为TDC（Town Distribution Center）、FDC（Forward Distribution Center）、RDC（Regional Distribution Center）、miniRDC、中心仓、卫星仓、协同仓、商家仓等。

有了支援关系网络，还需要知道往各节点补货的成本，哪一种最划算，它也提供了往哪儿补的基础。

那么补多少的基础是什么？显然是商品库存，需要知道全网的库存情况。

什么时候补的基础是什么？这个主要取决于到货时长，及预计用户收货时间。到货时间越长，时效要求越高，就越需要尽早补货。

还有一问题是能不能补？这属于一个约束，主要取决于风控决策和产能，有补货风险或产能不足，会影响补货计划。

综上，我们会看到，影响订单驱动补货计划的因素有很多，如何把它们有效地结合在一起，得出一个最优解呢？早期的时候，团队采用的是基于固定规则的做法，但随着因素越来越多，复杂度呈指数上升，已经无法达到最优，每增加一个新维度，对于编码而言，也是痛苦的。实际上，它是一个多目标多约束的运筹学问题，通过算法可以解决。以订单、库存、支援网络为基础输入，综合考虑成本、时效、销量预测、风控决策、产能、供应商履约率等因素，构建订单驱动补货模型，通过它计算出最优补货路径，驱动采购或内配。

而订单驱动补货模型是订单履约决策中心的一部分，它对订单履约的各个环节，综合作出最优的决策。

特殊情况的处理：对于支援网络覆盖不到的情况，订单该如何处理？这个问题确实存在，但基于“暂停订单数据中心”建立的监控平台，会发现这些订单，并发出补货提醒。



每个采销人员都能看到自己维护的SKU的补货提醒，便于采取人工补货来弥补。在每次处理之后，系统会根据补货情况，不断完善支援网络的覆盖。支援关系网络如图5-16所示。

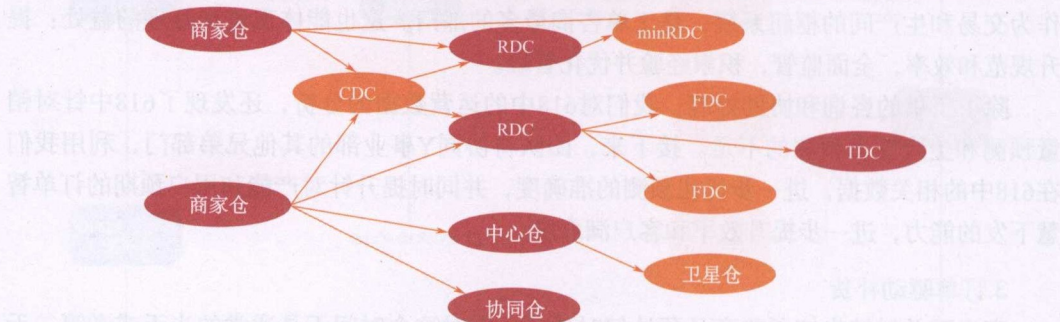


图 5-16 支援关系网络示例

从图5-16中可以看到支援关系是多级的，按照高级别到低级别的顺序，方向是单向的。就中小件而言，从供应商的商家仓开始，先到CDC（Central Distribution Center），再由CDC到RDC，从RDC再往下可以到miniRDC；也可以到FDC，或者到TDC。而大家电有所不同，它的支援网相对简单，只有中心仓和卫星仓之分，从商家仓到中心仓，再由中心仓到卫星仓。从支援关系上看，父节点到子节点，是一对多的，例如一个RDC可以支援多个FDC。

4. 大促履约策略

618期间，由于订单量的暴增，会造成产能不足的问题。以往的做法通常是把订单积压在库房或站点，待库房缓解之后再生产，但这样会造成用户体验不佳，订单取消率上升。

虽然我们在“运营决策分析平台”里能够看到各库房的产能情况，但在短时间内，运营人员是无法做到快速响应调配的。因此，我们在“订单履约决策中心”加入了产能动态调配，均衡负载的模型。

订单履约决策中心会根据各仓的实时产能负载情况支援关系网络，综合考虑成本、时效、销量预测、风控等因素，来动态调配产能。均衡负载会把一部分订单拉到其他相对空闲的支援仓来生产，同时也会考虑配送站点的负载情况，控制生产节奏，以保证用户体验，控制每个仓的生产节奏。

5.1.5 618系统质量保障

1. 618，活多资源少，质量如何保障

618是一个大日子，更是一个忙碌的日子。

对于用户来说，618每年从五月就开始了，一直延续到618达到顶峰。对于测试团队来说，618从二、三月份就已经开始了。根据项目提测的情况，这里把618分为了3个阶段：2~4月，预备期阶段；4~5月，爆发期阶段；5~6月，突发期阶段。

在预备期，都会有一些比较大的项目，为了618备战做准备。比如一般跟商品、补货、订单以及某些大的供应商进行协同的项目。这个时期的项目都是公司级或各个事业部的重

要项目，是早有计划的，所以测试人员一般也是可以满足的。

到了4月，进入爆发期，这时候第一阶段的项目还在进行中，但是有些事业部还是会新增很多与618相关的紧急项目、紧急需求。比如有些商品618要参加活动，或者要准备618之后的售后服务，一些商品、售后、备件库、退货等系统的紧急需求或项目就爆发了。这时大部分测试人员都已在项目中，但团队还是会预留一小部分人作为日常维护及系统紧急救援工作。不过预留的人数还是远远不够，按照正常的工作方式是无法应对测试需求的爆发的。这时团队就会通过自动化技术、优先级策略、人员统筹安排方法战略等来应对。

到了6月，一般的项目和需求是不允许上线的；一旦有上线都是紧急的需求和问题修复。为了应对突发阶段，大家都需要24h待命。

对于测试人员来说，618虽然分为三个阶段，但最有挑战性的还是第二个阶段。因为第二个阶段会爆发出各种情况，尤其是项目冲突、多项目并发、测试资源严重不足。但解决这几个问题，主要还是需要依靠技术来解决。

2.技术先行：速度与质量

618期间最大的问题就是时间紧、任务重，质量部的同事随时都处于紧张忙碌的状态之中，但是对于质量的要求自始至终都要保证高标准。在这种情况下，技术先行是测试与质量管理部的重头戏。

从整体流程上，把技术贯穿全程，做到整体质量流程的可控可追溯，同时提升各个环节的工作效能。团队将技术工作的落地分为三个方面：流程控制、持续集成、测试自动化，三方面的技术相辅相成，互为依存，如图5-17所示。



图5-17 技术工作的落地



Leon：供应链履约研发部专家。
tao：供应链协同研发部专家。
Arthur：供应链库存研发部智能补货负责人、领域专家，负责京东智能补货的管理、研发、维护等工作。

5.2 智慧物流

5.2.1 精准履约与运营数据服务

1. 履约时效及运营数据管理系统的整体功能

京东履约时效及运营数据服务平台是为京东物流提供时效计算、运营数据分析与决策支持的关键平台，其整体架构如图5-18所示，主要包括对外服务层、履约计算层和数据管理层三个层次。

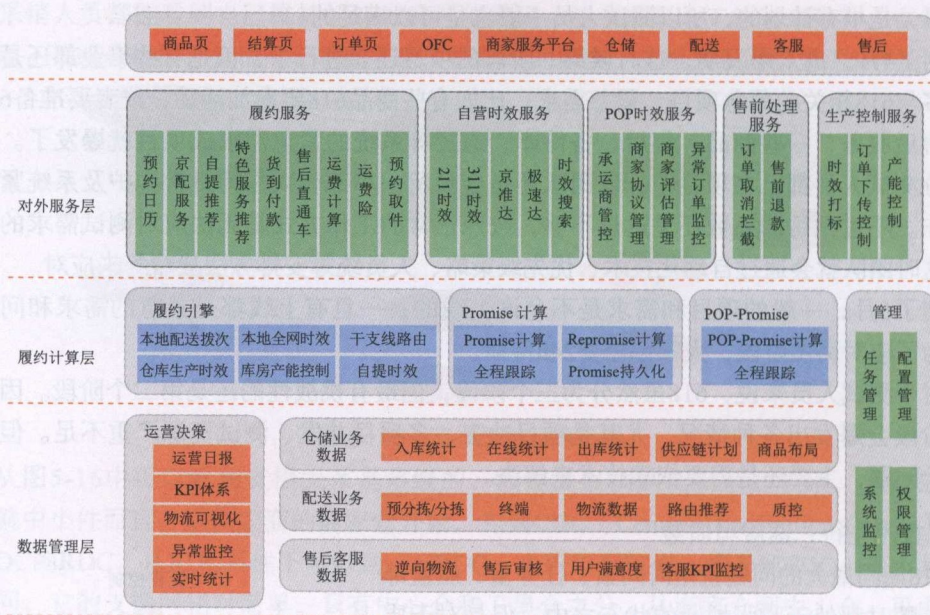


图5-18 履约时效及运营数据服务平台整体架构

其中，对外服务层通过商品页、结算页、订单页等系统，为用户提供预约日历、自提推荐等履约服务，211（一日两送）、311（一日三送）、京准达（准时送达）、极速达、定时达等时效服务，第三方商家、承运商管理等POP时效服务，订单取消拦截、售前退款等售前处理服务，以及时效打标、订单下传控制和产能控制等生产控制服务。

履约计算层提供核心的计算服务，主要包括核心的履约引擎和对应的Promise计算和POP-Promise计算服务。履约引擎通过时效波次、配送时效方案、仓库时效方案等，组成全国网状路由覆盖，对订单预约、下传、生产、配送各环节的时效进行计算。Promise和POP-Promise在履约引擎提供的基础计算之上，提供订单全链路的时效计算、全程跟踪、时效再计算以及持久化的服务。

最后是运营数据管理与分析层，包括仓储、配送和售后、客服等在内的整个运营系统的各种类型的数据，涵盖入库、出库商品统计信息，供应链、商品布局信息，订单预分拣/分拣信息，物流数据、逆向物流、售后审核、用户满意度等，具有系统进行存储、管理，并进行分析以提供运营日报、KPI体系、实时统计、物流可视化以及异常监控等决策支持。

2. “约”为人先——系统的特性展示

以211限时达、京准达为标志的用户时效服务，作为京东重要品牌之一，已成为电商发展中时效创新、改革的重要里程碑。快速、准确履约，是影响客户体验的关键因素之一，同时也是生产运营精细化管理中非常重要一环。以履约时效体系为核心，京东履约时效及运营数据服务平台提供了多种业务及关键技术的支撑，使得京东在履约时效服务方面可以做到步步为先。

◆ **基于模式的细粒度用户时效。**履约时效体系，从下单前引导用户时效预期开始，贯

穿订单全流程，对订单生产节奏进行控制，并可根据仓、配的变化，实时给予用户履约时效引导，保证订单的有序排产。

如图5-19所示，履约时效体系是一种计划，前台用户感知的履约与实际生产联动，形成完整的时效体系。用户前台选择的时效，直接影响仓库、配送、干支线路由、终端作业生产；实际运营生产节奏，直接反映给前台用户时效的感知。

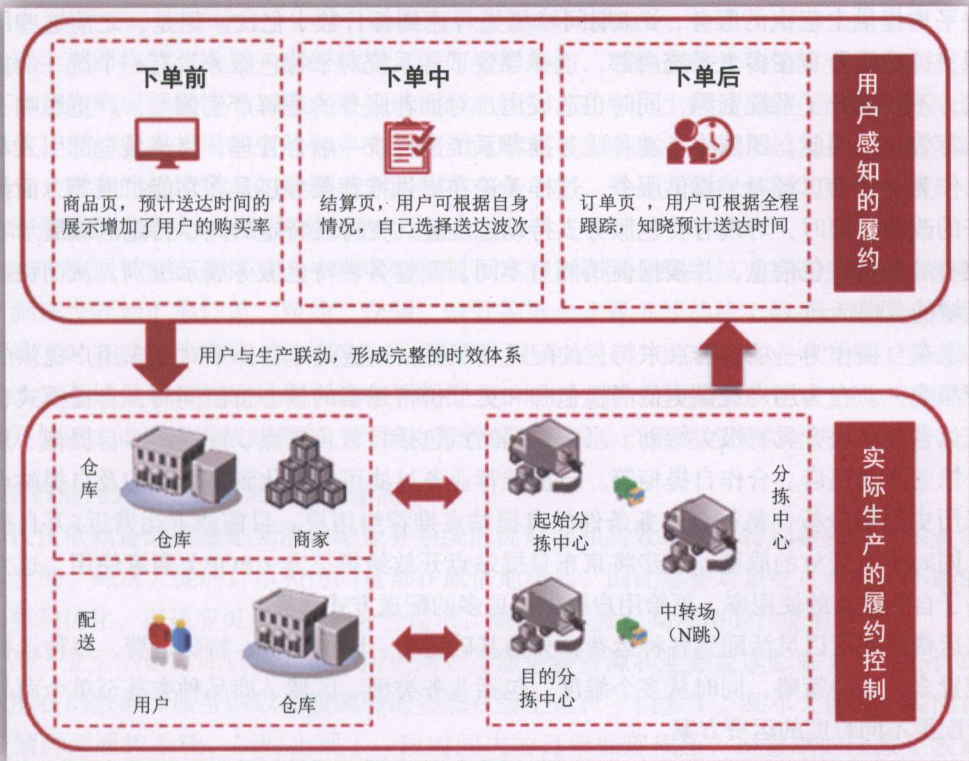


图5-19 用户与生产联动的时效体系

除了用户熟知的211限时达时效，京东还提供了包括311、极速达、定时达等在内的多种细粒度的时效类型，以期为用户提供更精细化的配送服务。因此，系统引入了“时效模式”概念，系统可根据实际业务需求，灵活定义时效模式，可以从商品类型、运输类型、业务类型等多个角度进行定义，如干线、生啤、生鲜业务的正向、逆向生产流程等，从而支持新业务的快速接入。

◆ **多维度、可定制的生产履约标准。**首先，系统实现了履约引擎的服务化，可提供多维度定制化履约服务，根据Promise给定的最晚出库时间进行排产安排，取代先前的JIT波次控制，增强集团对现场生产管控，提升管控水平。履约引擎支持从多个维度，包括仓储、地址、机构控制、特殊时段、多跳、特殊业务类型、不同运输类型等单仓、特定路由（区域）、特殊时段等纬度，实现对生产环节的精细控制。

同时，系统基于多节点路由控制，实现生产流程的可定制化，包括订单转移控制、生



产全流程计划、生产预约、时效查询、预计时效推送、承诺时效推送、履约计划持久化等，从而大大提升了新业务的开展速度。

◆ **统一、开放、多样化的特色服务系统。**广义上的特色服务系统包括商品详情页的服务类型推荐、自提服务推荐、运费计算服务等。目前京东已经开通4大类50多个的特色服务类型，例如购立返、京尊达、送装一体、家电清洗等。特色服务分为时效类、支付类、服务类等，以ICON形式展示，在给用户呈现时效物流基础服务时还添加了其他特色服务。系统平均提供上亿次的服务，618期间峰值更可达到每日数十亿次。但是，之前这些服务类型支持功能分布在多个系统内部，这样导致了各系统对于特色服务没有一个统一的推荐模式，不仅浪费了系统资源，同时也造成用户对推荐服务的理解产生偏差，严重影响了服务推荐效果。因此，团队将各端各服务推荐系统进行统一融合管理，将流量全部引入本系统，作为统一的口径对外提供服务，这样无论在提供推荐服务还是在功能扩展等方面都有很好的改善。同时，对现有特色服务支持动态配置，支持在特定时间、特定区域展示本区域独有的促销特色信息，并根据促销侧重不同，调整各种特色服务展示级别，从而快速满足区域的营销活动。

京东自提作为一项极具京东特色的配送类服务，一直是运营体系对京东用户提供的核心产品之一。在为用户提供更低门槛包邮和更低价格运费的核心价值同时，自提模式也对京东运营效率提升具有极大帮助。京东自提站点包括自营自提点、配送站、自提柜、京东派、第三方便民店、合作自提柜等。自提推荐业务过滤可以细化到每个SKU及自提站点，基于历史数据分析，将符合业务条件的自提站点推荐给用户。目前线上运营近5万自提站点。同时秉承开放的原则，逐步将京东自提站点开放给第三方（POP）商家使用，这大大提高了自提站点的使用率，更给用户提供了更多的配送方式选择。

运费系统可以灵活配置各种运费，分为基础运费、附加运费、特殊运费，每种运费又可配置多种运费策略，同时从多个维度，包括业务类型、区域、商品种类甚至单个商品的维度配置不同粒度的运费方案。

◆ **快速响应的订单取消调度系统。**经过多年的发展，系统的取消快退平台以订单中心为基础，以货物拦截和财务退款为核心支撑，以拦截组件、退款组件、数据集市、用户中心、订单中心等公共组件和服务为重要辅助，搭建起一套便于复用、易于扩展且符合快速退款业务特性的订单取消退款流程。系统支持多种业务类型，支持多个环节的订单拦截，自营订单、POP订单、虚拟订单都逐步纳入到了取消快退平台范围，实现了订单自动审核退款，节省逆向物流损失的同时提高了用户的购物体验，缩短了退款时间。同时，取消订单、物流拦截、退款彻底解耦，做到物流、资金流清晰、顺畅，恶意订单拦截快速响应。取消快退平台将订单取消功能统一管理，有效地协调了各系统之间的工作，提高了订单取消效率、取消成功率，对商品库存的利用率得到有效利用，同是也对商品资金的安全提供了有力保障。

◆ **精准配送的三方商家履约系统（POP时效服务）。**POP时效服务作为三方商家物流数据中枢和规则制定者的角色，主要功能是接入履约及物流相关的数据，并对其分析、监控，对商家、快递公司进行能力、口碑等级划分，并在此基础上制订快递公司可用。对履约好的

快递公司、商家，给予优先推荐使用或搜索、履约考核支持以及各种履约数据支持服务，提高京东对于POP商家与社会快递的可控性，并且提升商家与社会快递的服务质量。

为了保证第三方商家的送货配送时效，京东依托海量的信息数据，包括日均千万量级的发货数据、百家左右承运商信息，覆盖全国的亿级路由数据等，推算出这个商品在客户所选路由上的时效信息。系统将计算出来的信息提供给下游系统，从而达到控制订单下传的目的，保证了承运商在预定的时间内准时将货物运送到用户手中。

◆ **面向实时的数据统计分析。**随着京东业务的不断拓展，订单类型越来越丰富，也使得仓库生产面临着一次又一次的挑战。例如生鲜品类，对配送时效和保鲜要求非常高，基本上不允许出现积压情况，因此需要进行产能控制，在库房生鲜订单的生产能力还未到达极限前，提前发现并限制生鲜订单的下传。而随着京东线上业务越来越复杂，系统不断地做细分，所有的监控需求的数据一方面分布在多个库，另一方面数量极其巨大，需要系统面向海量数据满足实时统计与分析的需求。

实时数据平台正是针对这一需求而定制开发的，从刚上线的仅支持部分配送和履约数据，到现今开始汇集订单、履约、仓储、退款等近十条业务线的线上数据，提供业务线全流程的实时监控统计数据，日处理数据量达到了10TB级，并随着京东业务发展、业务线逐渐接入，存储量和日处理量每天都在不断扩大。同时，随着数据头条正式上线，实时数据平台开始正式走向高层战略管理、区域决策管理、一线生产管理的探索道路。

3.效能为先——技术架构及创新展示

仅仅依赖备战措施是无法保障业务系统的高性能和高效率的，特别是随着京东业务的迅猛发展，每次大促的订单和访问量都在成倍地增加，因此需要对系统进行持续不断的架构调整和优化，以适应灵活多变的业务需求，满足高并发、高可用的性能需求。

◆ **基于业务的数据库拆分。**由于历史原因（服务器资源和业务急速膨胀），核心订单时效库所在的数据库服务器以比较臃肿的姿态在线上运行了很多年。某年大促时，瞬间订单洪峰倾泻到履约系统，当时出现了一段时间内的订单处理积压，很直观的现象是数据库服务器CPU使用率居高不下。经过慢SQL分析发现，那一刻同时也有一些非核心业务的大SQL在执行任务，这些SQL属于复杂（包含大表的分组、排序）的查询类SQL，而核心业务主要是处理订单信息的写入输入高并发写入类SQL。数据库的拆分势在必行。

团队主要从两个层面进行拆分：

- ◆ 该服务器上的非核心库全部原封不动地迁移到其他服务器上。
- ◆ 核心时效库中的非核心表也拆分到其他服务器的新库上。核心库只应付海量订单信息写入，使得后续TPS的优化变得更加有针对性。

◆ **网络I/O优化。**先前由于缓存数据结构设计得不合理，导致差点将分布式缓存集群的千兆网卡打满，故而后来进行了有针对性的缓存数据结构的优化。按照峰值调用量、峰值数据传输流量大小等评估缓存结构设计是否合理，在调用次数与数据冗余上进行综合考虑和取舍，同时将分布式缓存集群也按业务进行了合理的分离。

现有分布式缓存集群加入新数据结构时需要考虑集群的容量、QPS峰值限制、带宽等因素，需要扩容的则扩容，提前将风险扼杀。



磁盘I/O优化。主要体现在应用日志的写入量与写入阻塞上。曾有个应用在促销过程中，在短短几分钟内就往磁盘上写入了上百GB的日志。经排查发现大量是行为式的日志导致。也曾发现过某些应用在流量飙升时会出现处理能力下降的问题，经过Jstack抓取线程堆栈信息发现，发现大量的日志写入线程是BLOCKED状态，服务器也出现磁盘写入队列大量排队现象。

对于以上描述的现象，团队做了几点优化：

- ◆ 从根本上优化应用日志输出（精简、有效、每个请求的日志都绑定唯一的标志，方便排查问题），真正做到打印日志是一门艺术。

- ◆ 日志按业务或包分文件，配置不同的日志级别和滚动策略，控制所有日志文件总大小不能超过服务器磁盘空间大小的70%。

- ◆ 根据实际情况调整日志工具的BufferSize，在适当场景下使用异步日志。

多级缓存策略。为应对618等高流量的特殊时期，并且为了进一步提升服务的高并发、高性能，团队采用多级缓存的方式对数据进行处理，同时针对不同的业务应用，采用不同的缓存策略。例如，对于京配服务、特色服务推荐等响应时间要求较高，而数据一致性要求相对较低的应用而言，采用了分布式缓存加本地热点缓存的方式。对于分布式缓存，采取的是定期全量缓存更新策略，数据首先写入数据库系统，每天在固定的时间将数据同步到分布式缓存系统。同时针对本地缓存，采用热点数据的方式，针对某些秒杀活动，基于历史数据或者其他信息进行本地缓存预加载的方式。并且针对不同的业务数据，设定合理的本地缓存的时效时长，从而更有效地提升系统的访问效率，进而提高系统性能。

平衡海量数据和实时计算的架构。面对TB级数据量的统计查询和明细查询，要求毫秒级到秒级以下的查询性能，这样一个近似矛盾的苛刻要求，实时数据分析平台选择了一个相对平衡的实现方式：

- ◆ 选用ElasticSearch作为数据落地的存储引擎，借助ElasticSearch高效的读写性能，完成实时数据模型的存储和查询功能。

- ◆ 通过Kafka实时接收线上业务数据，通过实时数据引擎，根据预先设定的实时数据模型进行数据实时处理，并将实时处理明细落地到ElasticSearch。

- ◆ 通过调度系统汇总查询统计需求，由调度系统转发用户查询请求，并提供动态查询参数，这样能够控制非常消耗ElasticSearch集群的查询请求，保证ElasticSearch集群的资源消耗是可控的，又能够做到满足用户复杂的查询统计需求。

- ◆ 对于实时性能没有非常高的查询统计需求，增加数据中间缓冲层，将固化的统计需求结果定时刷新到数据缓冲层，这样就可以将ElasticSearch集群压力转移到数据缓冲层，缓解ElasticSearch集群查询压力的同时，提高请求的吞吐能力。

自定义的业务监控系统。业务流量到一定程度之后，系统的各维度的监控尤为重要，可以帮助我们缩短排查、定位问题的时间，甚至可以帮助预警风险。目前京东公司级监控系统做到的监控包含：网络层面、服务器层面、应用层面、JVM层面、数据库层面。但是各个业务系统也有可能对自身的业务数据的正确性也比较敏感，针对此类需求，公司级监控系统就很难做到深入所有的业务系统，所以团队自主研发了一套表数据级别的监控系统

FDM（Fulfillment Data Monitor）。

FDM系统可以针对不同的表配置不同的监控任务：异常数据扫描SQL、异常判定条件、任务执行计划、报警策略等。

经过多次618的洗礼，京东履约时效及运营数据服务平台已经逐渐成长为高并发、高可用、高性能，支持全面履约时效体系的计算服务平台和运营数据分析处理平台，并随着京东业务的不断发展而不断迭代和发展。未来，平台还会有一系列的优化计划，相信系统会更加的壮大，更能适应京东飞速的发展。

5.2.2 智慧仓储

1. 仓储大动脉——DTC仓储数据路由中心

数据交换中心系统是京东仓储系统与上游ERP各系统交互的通道。功能包括：平滑仓库接收的数据，避免高峰流量对仓库系统的冲击；京东各个仓储系统的数据路由；对不同API进行协议转换；兼容可能出现的各种异常；方便运营人员使用等。

从2013年上线开始，陆续接入京东几乎所有的仓库和生产业务，数据的处理能力由几百万的量级，迅速攀升至百亿级别，系统愈来愈健壮，支撑订单履约，助力京东物流业务快速发展。

随着京东的发展，物流系统尤其是仓储系统需要的不再是一个简单的入口，而是一个功能更加强大的系统来支撑。于是，数据交换中心应运而生，如图5-20所示。新的数据交换中心系统更加平台化，所使用的技术的实现方案更加灵活与健壮。

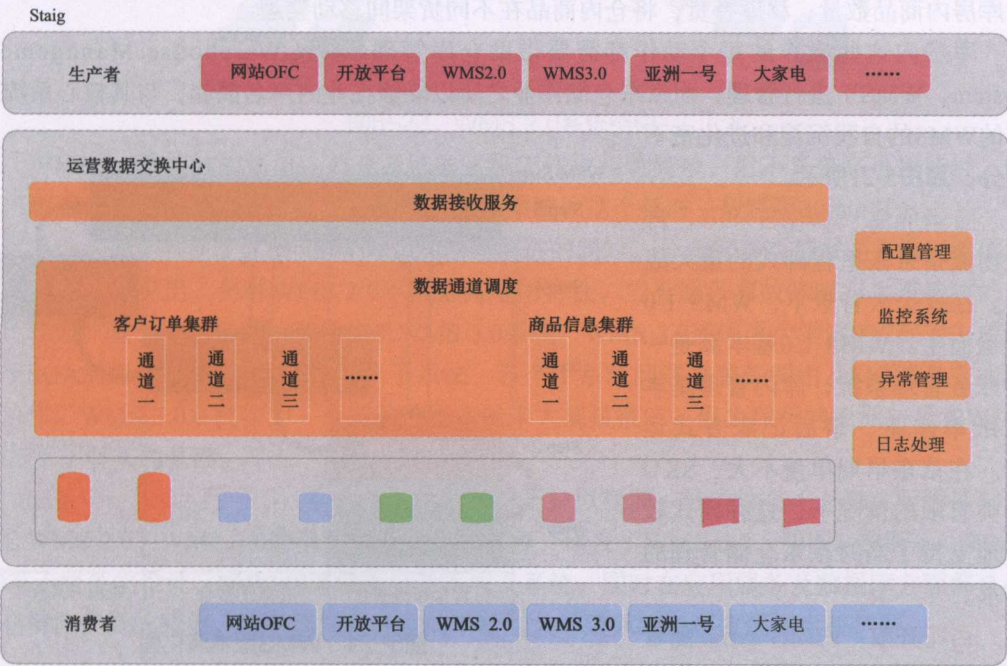


图5-20 数据交换中心



◆ **缓存**。业务发展迅速，数据交互在短时间内会达到可以冲垮业务系统的量。异步方案降低系统依赖，数据持久化降低缓冲目标系统的压力。上游来的数据放入JMQ中，称为数据通道。

◆ **路由**。路由配置手段和缓存方案升级。

◆ **协议**。支持CXF、RESTful、JSF、JMQ等多种协议。

◆ **健壮性**。冷、热任务使用不同的持久化策略和多级别重试机制。

◆ **历史数据查询**。数据报文日志放入HBase长久保存。业务数据可追溯。

◆ **维护**。模块功能清晰，便于维护扩展，多角色参与项目，做到数据及流程的透明。

数据交换中心作为京东仓储业务的大动脉，必须要保证准确的同时，具备大数据量、快速处理、高可用及分布扩展的功能。

2.WMS玄武之蜕变记

如今电商市场格局未定，风云变幻，各大网站为抢占客户资源，均使出浑身解数；从PC到移动，从价格到品类，从促销到广告……然而，天下武功，无坚不摧，唯快不破。

在一番比拼以后，各公司都渐悟出一个道理：得物流者得天下！可物流内功的修炼岂是易事？对于一家电商公司来说，欲在物流的角逐中领先对手，必须具备两大功能：一为快速准确的仓储生产，二为高效发达的配送网络。这里重点介绍仓储，仓储生产主要分为如下几个环节：验收，验收人员对送达库房的商品检查其数量和质量；上架，将验收到达库房的商品，库房人员摆放到仓库内的货架上；拣货，根据订单将需要的商品从货架取下；复核，核对订单、商品、发票等信息；打包，将商品打包或装箱进行发货；盘点，核对库房内商品数量；移库补货，将仓内商品在不同货架间移动整理。

库房内这些操作的每个动作都需要借助仓库管理系统（Warehouse Management System，WMS）进行管理，而京东仓储作业之所以能够在业内声名鹊起，与其核心系统之一的WMS的自我沉淀和进化密不可分，如图5-21所示。

应运而生：WMS 1.0。自建物流是京东里程碑式的重大战略。在这个大背景下，WMS 1.0应运而生。WMS 1.0基本算是一个账务管理系统，库内管理基本以纸单作业，事后记账方式运营。在京东早期单量不大、SKU种类有限的情况下，这种模式较好地支持了当时京东仓储管理的需求。

百花齐放：WMS 2.0。随着京东订单量指数级的增长，自营仓库越建越多，仓内管理的SKU也越来越多，记账式的WMS 1.0已满足不了库内作业管理

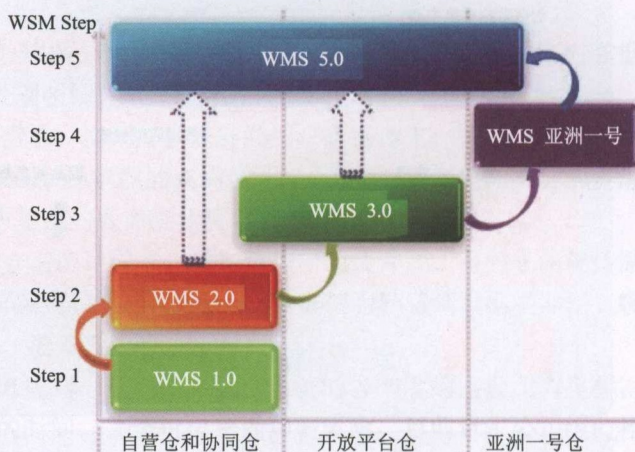


图5-21 WMS版本演化图

的及时性和准确性要求。为提升仓库运营的效率，急需一套可满足各个品类、功能更加齐全，适应全国各品类仓库的标准系统。

另外，在IDC集中部署情况下，由于仓库到IDC网络的不稳定造成的系统卡顿在1.0时代也是一个问题，不至于产生太大问题，但在全流程作业管理方式下，就会成为一个大问题；因此2.0采用了分布式部署方式——将WMS系统部署在各个仓内，以规避网络问题。

WMS 2.0是一个典型的胖客户端模式，客户端直连数据库，主要的业务逻辑写在客户端，系统结构如图5-22所示。

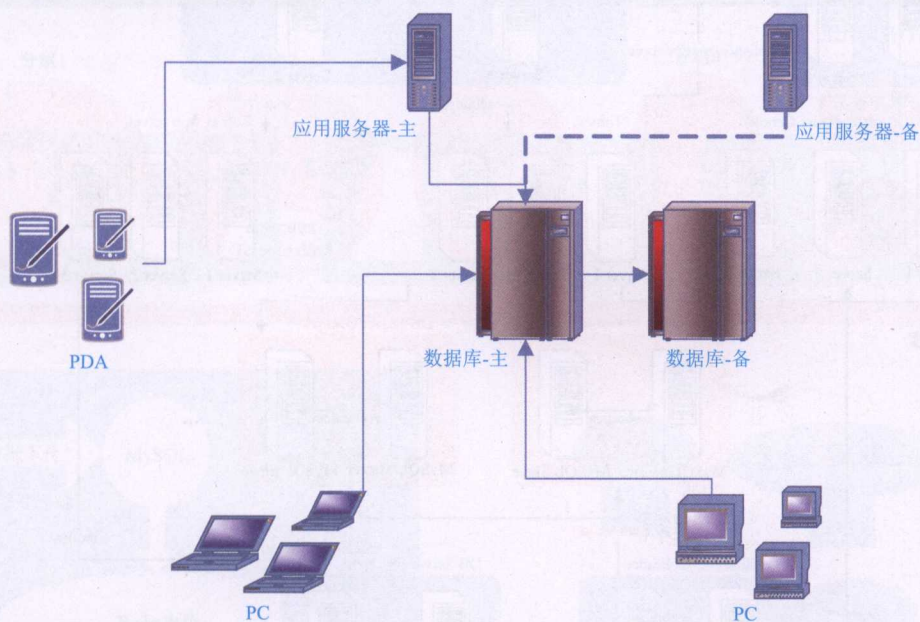
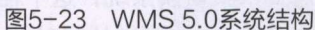
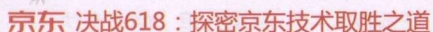


图5-22 WMS 2.0系统结构

WMS 2.0系统结构简单，单套系统的运维工作也相对简单，但不具备高可用特性，线上出现服务器硬件故障等情况下需要手工切换系统，这个耗时一般需要30min以上。

海纳百川：WMS 3.0。在WMS 2.0快速演进的同时，分仓部署所带来的系统运维困难的问题越来越突出，另外WMS 2.0不具备高可用特性，当有服务器故障时对生产造成的影响比较大。基于这些问题，开始构思WMS 3.0系统。WMS 3.0系统再次尝试IDC部署方式，基于SOA的思想将WMS拆分为各个子系统，各个子系统间通过服务调用及MQ消息方式通信协作。WMS 3.0系统作为一个实验性版本虽未大规模推广，但为后续的亚洲一号和WMS 5.0打下了坚实的基础。

破茧蜕变：WMS 5.0。在仓储开放的背景下，以亚洲一号中件仓（上海）版为基础，结合WMS 2.0、WMS 3.0的系统特性及业务流程，研发了WMS 5.0，如图5-23所示。WMS 5.0基于服务化设计，将WMS系统拆分为不同子系统，同时在应用服务及数据库方面都具备按仓横向扩展的能力，避免因为业务量的不断增大导致系统出现瓶颈却无法扩展。



◆ WMS 2.0的一个优点是各仓独立部署，互不影响；这样线上系统出现Bug（特定数据条件下）导致的系统不可用，只会影响到一个仓，问题范围可控。因WMS 3.0是共享服务资源，测试过程中未发现的Bug，在某仓的未预期操作引发的情况下，可能导致其他仓也会受累。

因此，WMS 5.0采用IDC加园区部署方式——都是多仓共用一套系统，但也具备按仓隔离的特性，如图5-24所示。

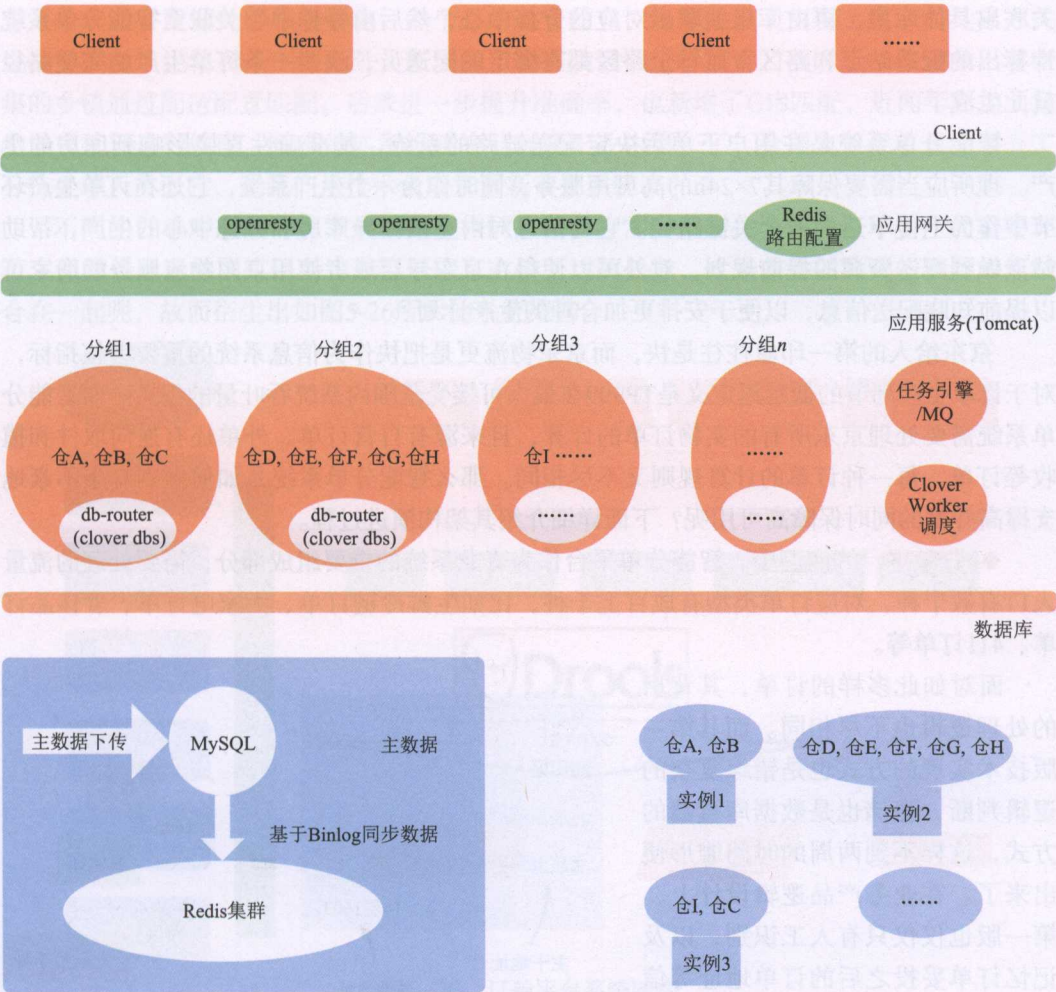


图5-24 IDC加园区部署方式

在数据库及应用服务实例层面都是按仓设定分组，在请求的上下文中含有仓库编号，基于仓库编号实现应用实例及数据库实例的路由，这样可实现在特定Bug出现时更改路由设置，将与问题仓共享应用实例的其他仓切换到其他分组，避免问题影响范围的扩大化，减少线上故障处理的压力。

5.2.3 智能配送

1.海量地址的狂欢——智能分单系统的蜕变之路

7×24h不间断低延迟服务。智能分单系统的作用是根据收货地址、商家属性、订单类型等计算出末端配送信息的系统，即配送的站点以及站点对应的路区。

如果觉得上述定义有些抽象，那么我们开启一段订单生产之旅。当您在京东网站下单，提交订单计算完成之后，后台系统要迅速地根据下单商品对应的SKU和客单所在城市



关联出具体库房，再由库房关联出对应的分拣中心，然后由分拣中心关联至智能分单系统计算出的配送站点和路区，而每个路区都有绑定的配送员，故而一条订单生产的完整路径就此生成了。

智能分单系统是在用户下单后决定配送链路的系统，其准确性直接影响到库房的生 产。理所应当需要保障其7×24h的高可用服务。同时作为一个生产系统，它还在订单生产环节中作为上传下达的一个关键路径。它的信息对内直接指导库房和分拣中心的生产，帮助站点做到配送资源的提前规划，对外可以使得在京东开店或者使用京东物流服务的商家可以提前知晓配送信息，以便于安排更加合理的生产计划。

京东给人的第一印象往往是快，而京东物流更是把快作为信息系统的重要考核指标，对于订单生产环节的低延迟定义是TP999在最大可接受范围内系统吞吐量的拐点。而智能分单系统需要处理京东所有的实物订单的计算，且来源有自营订单、外单还有逆向取件和揽收等订单，每一种订单的计算规则又不尽相同。那么智能分单系统是如何做到有条不紊地支撑高并发的同时保障高可用呢？下面详细介绍其架构演进过程。

◆ **分单平台的血泪史。**智能分单平台作为青龙系统的重要组成部分，需要处理的流量入口有数十种，对应订单类型有成百上千种，比如生鲜冷链订单、大家电订单、奢侈品订单、411订单等。

面对如此多样的订单，其背后的处理逻辑也不尽相同。而其第一版技术实现的方式也是错综复杂的逻辑判断，存储也是数据库直给的方式，这样不到两周的时间雏形便出来了。在业务产品逻辑设计上，第一版也仅仅只有人工识别，以及记忆订单妥投之后的订单地址等信息（图5-25）。上线后没多久，京东迅猛的发展以及单量阶段性的翻

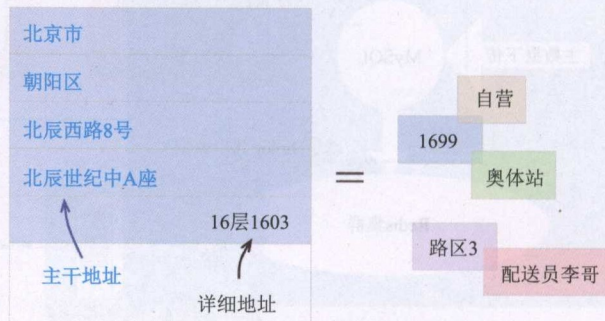


图5-25 第一版业务处理逻辑

倍，其分单功能也愈加繁杂，数据量越来越大，系统维护成本自然越来越高。但分单的准确率并不高，人工分拣人员需求量越来越大，且培训成本很高。是时候需要进行一次革命了。

动员之后，团队便没日没夜地封闭起来了。在青龙封闭的那段日子，公司特意又租了北辰世纪中心的一层楼，并且在附近订好了长期的宾馆，那会儿大家脑子里只有上线，只有技术和各种业务的融合。如此大规模的SOA物流配送系统在国内实属首创，1.0系统两三个月之后便试上线了。如今系统已经发展到了6.0版本。多年以后再聚首时，每次说起来当时的种种依然激情澎湃。如今虽然已经搬到了亦庄，好几年没吃到北辰的加班餐了，但每次路过北辰楼下的加州牛肉面，看到橱窗依然热气腾腾，心里总是莫名的温暖。

到了2.0阶段，智能分单系统一改错综复杂的逻辑判断，运用规则引擎+设计模式来重新实现业务。与此同时，也开始从头梳理所有的订单匹配逻辑，重新定义业务。缓存化、

容灾处理、并发设计等相继孵化，一个月的通宵达旦，终于迎来了灰度上线。而计算方式也在妥投地址记忆和人工匹配外，新增加了根据关键词进行搜索，而对于一些订单不够密集的乡镇通过配送配置匹配。后来进一步提升准确率，也新增了GIS匹配，近两年新加了自然语言处理进行匹配。现在的匹配准确率已经属于业内第一了，全国也仅仅有几十名人工分拣处理人员而已，让我们从技术角度来重新审视系统。

不同的订单对应的计算方式不完全相同，且会经常发生变化。作为订单生产的重要环节，保障低延迟的同时还要完成订单信息上传下达的任务，这么多看似冲突的特性如何结合在一起呢，故而衍生出如图5-26所示的系统架构。

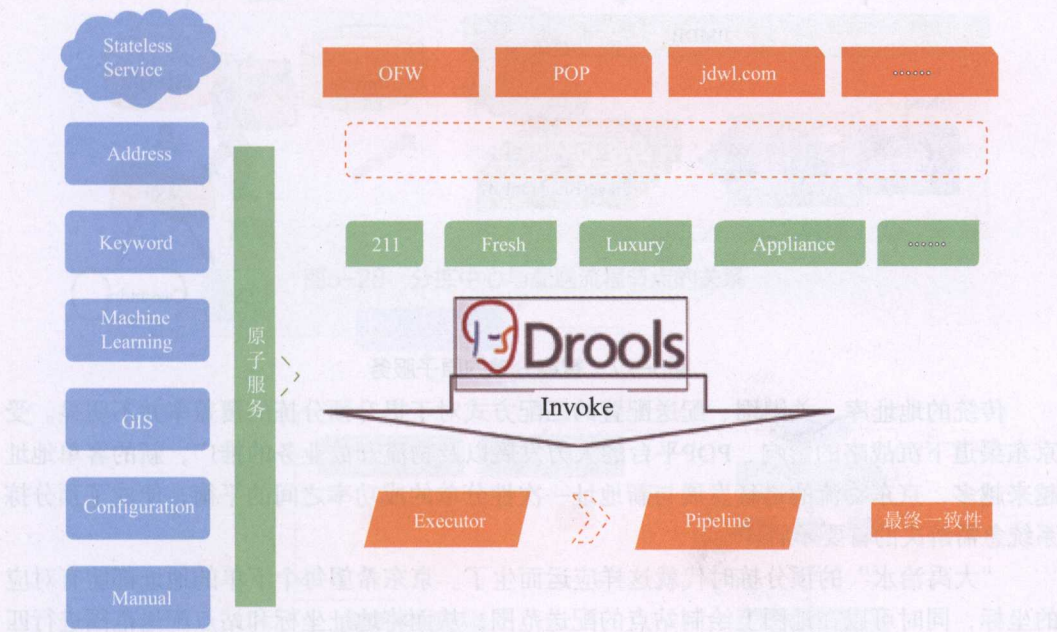


图5-26 订单平台系统架构

由于计算方式有很多种，故而把每种计算方式封装成一个一个的原子服务，如地址库服务、关键字检索服务等。而每种订单类型对于原子服务的计算诉求又有些差异，故而需要把所有订单类型梳理成一条一条清晰的订单规则。而驱动订单规则与原子服务计算的，正是规则引擎（Rule Engine）。经过性能测试的比较，我们最终选择了JBoss的Drools 6.5.0版本。Drools的语法清晰明了，case when then的简明形式把规则和原子服务有机地结合在一起。为了更加便于Drools和Java的交互，团队还把所有的原子服务通过Invoker来实现。当计算出结果后，还需要存储、生产MQ、发送全程跟踪等，为了保证低延迟的特性，故而采用了结果数据的最终一致性策略。

拆分规则和原子服务后，规则的定义可以由产品人员灵活地配置，再选择与之匹配的原子服务。当有一种新的订单产生时，产品人员完全可以独立完成，实现了代码“零”改动即响应业务需求变化。

◆ **极致的用户体验。**极致的用户体验是京东追求的永恒目标，并为此乐此不疲。保障低延迟服务的策略有很多，其中重要的一点是缓存，京东有着海量的地址数据，且对检索



的性能要求很高。

如图5-27所示，对于智能分单的原子的服务，一些常用配置数据一般会采用本地+集中式缓存的架构方式。而对于一些易变化且数据量很大的系统，一般会采用集中式缓存的方式来实现，如Redis、ES、Cassandra等。为了进一步提升性能，团队会把读分组进一步细化为不同的分组，用来支撑不同的业务场景。

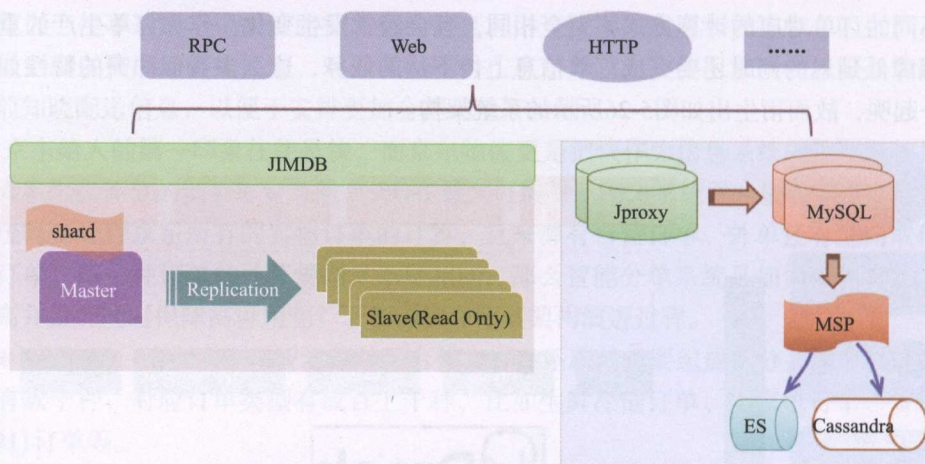


图5-27 智能分单的原子的服务

传统的地址库、关键词、配送配置的匹配方式对于提升预分拣的覆盖率并不明显。受京东渠道下沉战略的影响、POP平台的大力发展以及物流开放业务的推广，新的客单地址越来越多，京东物流的迅猛发展与新地址一次性分拣的成功率之间的平衡，便成了预分拣系统急需解决的首要矛盾。

“大禹治水”的预分拣时代就这样应运而生了。京东希望每个下单的地址都能有对应的坐标，同时可以在地图上绘制站点的配送范围，从而将地址坐标和站点配送范围进行匹配即可。

为了提升GIS的检索效率，通常还把站点的地理围栏缓存化，并先行计算每个站点范围的四个边界顶点坐标，便于更高效地检索。后期还改进采用了其他的投影算法，最大限度地避免球面到平面转换的失真，由此正式开启了数字化物流的新篇章。GIS技术本身存在不完美之处，针对GIS计算方式有一定程度损失精度的问题，除了优化GIS算法，团队还采用了自然语言处理来补充，进一步提升了自动预分拣的准确率，大大提升了物流配送的效率。

2. 仓储与配送站的纽带——分拣备战大事记

◆ **细化纽带——物流数据流之分拣中心。**分拣中心是物流配送的枢纽，连接仓储库房与终端配送站。所有库房的订单，从库房发出之后先集中到分拣中心，到分拣中心后，根据相应的路由规则，分配到不同的配送站，如图5-28所示为分拣中心与配送流程节点的关系。

分拣中心系统前端承接运单、预分拣、基础资料系统获取信息，后端连接终端及运输系统，处于整个配送体系的核心位置。分拣系统的稳定性也决定了整个京东配送系统的可用性。分拣中心从部署架构及数据库设计两方面入手提升系统的稳定性及可用性，在部署

架构方面，对一些大型分拣中心由IDC机房应用和本地机房应用共同提供服务，优先使用本地机房服务器，在本地机房不可用时会自动把故障转移到IDC机房应用上；在数据库设计方面，通过数据前置和多级缓存的方式提升数据库的读写性能和查询效率。分拣系统架构如图5-29所示。



图5-28 分拣中心与配送流程节点的关系

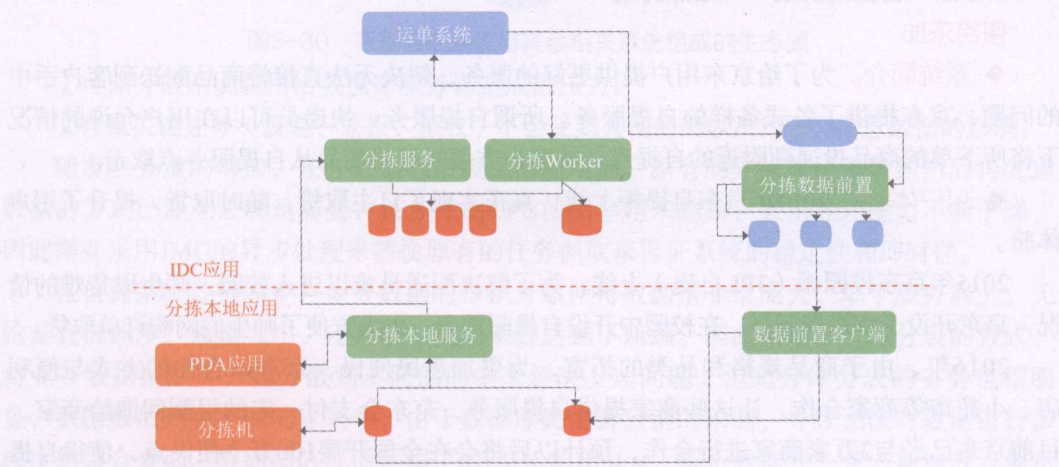


图5-29 分拣系统架构图

分拣中心架构分为两个区域，上半部分为IDC应用区，部署在京东的中心IDC机房，这部分应用为分布在全国各地所有的分拣中心作业提供服务；下半部分为分拣中心本地应用，部署在全国各地分拣中心本地，这部分应用只为本分拣中心服务。

IDC应用模块有三个，分别为分拣服务、分拣Worker、分拣数据前置。分拣服务负责接收本地的分拣服务请求，提供分拣实时报表服务；分拣Worker负责将分拣数据回传给运单服务，由运单服务对外提供全程跟踪相关信息；分拣数据前置负责前置待分拣运单数据，它会做一个路由，使每个分拣中心需要前置的数据量最小化。

分拣本地应用模块有三个：数据前置客户端负责将前置的运单数据储存到本地数据库内；分拣本地服务负责接收分拣请求，验证分拣目的地和分拣逻辑的正确性，并异步上传



分拣数据给IDC的分拣服务；PDA应用负责PDA操作界面的相关功能。

◆ **无人分拣。**京东物流昆山无人分拣中心是目前全球首个正式落成并运营成功的全程无人分拣中心。无人分拣中心最大的特点是从供包到装车，全流程无人操作，场内自动化设备覆盖率达到100%。目前无人分拣中心的分拣能力已经达到9000件/h，在同等场地规模和分拣货量的前提下，据测算每个场地可以节省180人力。同时，一线设备的操作效率和运营质量也得到显著提升，对比传统供包能力，自动供包台的效率提升了3倍。

分拣中心常用的业务操作有验货、分拣和发货，每个业务节点都是由工作人员参与并推进完成。实现分拣中心无人化，首先就需要打破传统的操作运营模式，需要完全利用数据信息去驱动业务进程，推动并协调自动化设备稳定有序生产，所以无人分拣中心需要一个“大脑”，全程控制智能化设备强健的“四肢”，因此DCS智能管控系统便应运而生。

DCS智能管控系统是由京东物流集运研发部自主研发的定制化、智能化的设备管控系统，其中自动分包含拣机调控、无人AGV搬运调度、RFID的信息处理等。全场所有任务指令均由DCS系统中枢管控，完成AGV无人搬运、车辆安全、避让、优先任务执行等工作，实现了全国首例分拣前后端无人AGV自动装车、卸车作业的操作。

3.最后一公里的挑战——站点终端

便民乐加

◆ **系统简介。**为了给京东用户提供更好的服务，解决无法直接将商品配送到客户手中的问题，京东提供了各式各样的自提服务。所谓自提服务：快递员可以在用户允许的情况下将所下单的商品投递到附近的自提点，待客户方便时可将商品从自提服务点取走。

◆ **用户体验。**2014年京东自提柜上线，真正实现了自主取货，随时取货，提升了用户体验。

2015年京东校园派（3PL自提）上线，为了解决配送员难以进入校园，师生取货难的情况，京东开设了京东校园派，在校园中开设自提配送点，大大方便了师生们网购订单取货。

2016年，由于商品规格和品类的拓宽，为更加惠民便民，京东以合作的方式与便利店、小超市等商家合作，让这些商家提供自提服务，京东会支付一定的报酬回馈给商家。目前京东已经与3万家商家进行合作，预计以后将会在全国开展100万个便民点，使得自提服务能覆盖到全国的各个地方。

2017年，京东为了让第三方商家订单也能享受到自提服务，自提柜以及便民点由只面向京东自营订单开放至社会快递公司。

到目前为止，京东自提系统与其他相关系统组成了强大的生态圈，保证了系统的安全性以及稳定性，如图5-30所示。

◆ **架构演进。**最初的自提系统仅仅服务于京东自提柜，数据库只有MySQL单库，其中包括基础数据（用户信息、站点信息等）、业务数据（订单信息）、促销日志数据以及任务数据。后来，随着便民点以及京东校园派的融合，基础数据量开始成倍地增长，随之带来的是业务量的增长。自提订单量也在迅速增长，数据库单服务器单库已经无法满足业务的需要，因此将这些库拆分成四个库，基础数据库、业务数据库、促销日志库以及任务库，并分别部署在不同的服务器上，这样做的好处有以下两点：

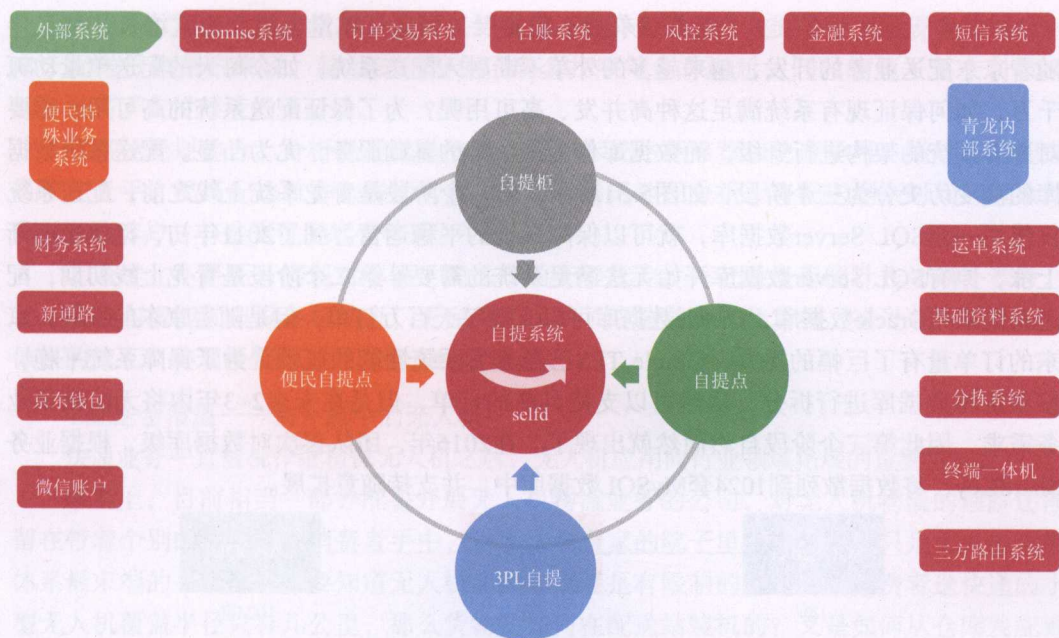


图5-30 京东自提系统与其他相关系统组成的生态圈

①可以分散系统调用给数据库服务器造成的压力。

②自提关键业务（投递、妥投、拒收）不会受到其他促销以及日志类信息数据的影响。

随着业务量的增加，任务数据的量级成倍地增加，原有的先抓取任务数据然后再处理数据的方式已经无法满足需要，任务库的CPU使用率越来越高，数据处理能力不断下降，因此需要采用JMQ的异步处理来替换原有的任务抓取来保证系统的稳定性和即时性。

在自提系统数据库中，业务数据的日积月累使得数据量非常庞大，单个服务器上，无论是查询速度、磁盘大小，还是CPU使用率都达到了瓶颈。因此采用了分库分表的方式，将业务数据按照拆分键分散到不同的库中来解决上述问题。但是分库分表的劣势也很明显，数据被散列在不同的库表中，由于数据库路由拆分键的限制，导致无法对数据进行多维度的整合查询。故而团队灵活运用MySQL的Binlog，解析数据并一条一条地放入消息队列中。在消费此队列的消息时，便可灵活地把数据加工到ElasticSearch的宽表中，如此便解决了大批量散列数据的多维度查询问题。通过不断地技术迭代优化，最担心的数据延迟问题也被控制在毫秒级范围内。

经过三年的演进，自提系统架构，降低了系统耦合性，将核心系统和非核心系统隔离，最大限度地保障了系统的高可用。

配送系统

◆ **系统介绍。**物流只是一个起步，当它变成资源平台后，能看到的东西就越来越多。作为物流系统的最后一公里，配送系统是电商环节最为核心的一环，近几年，电商爆发式的增长、订单量的成倍增长和客户对物流体验的极致要求，对电商物流系统来说都是极大的冲击和考验。而配送系统又是物流系统中唯一有机会跟客户面对面沟通的系统，因此配送系统的高可用、便捷性、高并发性会直接影响客户体验。



◆ **技术业务齐头并进。**随着京东的不断发展，配送订单量也呈几何式增长，并且伴随着京东配送业务的开发，越来越多的外单不断融入配送系统，如今每天的配送单量动辄千万，如何保证现有系统满足这种高并发、高可用呢？为了保证配送系统的高可用，需要对原有系统的架构进行升级，而数据库作为最核心的基础服务，尤为凸显。配送系统数据库的演变历史分为三个阶段，如图5-31所示。第一个阶段是青龙系统上线之前，配送系统仅使用一套SQL Server数据库，就可以保障系统的平稳运营。到了2011年初，订单量逐渐上涨，原有SQL Server数据库开始无法满足系统的需要。第二个阶段是青龙上线初期，配送系统采用Oracle数据库，Oracle数据库可以支撑每天百万订单，但是随着京东的发展，京东的订单量有了巨幅的提升，Oracle TPS已经成为系统性能的瓶颈。为了保障系统平稳，按系统将数据库进行拆分，虽然可以支持目前的订单，但是在未来2~3年内将无法满足业务需求，因此第三个阶段自然而然就出现了。在2016年，团队再次对数据库级，根据业务ShardKey，将数据散列到1024套MySQL数据库中，并支持随意扩展。



图5-31 配送数据库演变流程图

同时，为了保障系统的高并发、高可用，团队通过数据总线的方式，将数据MySQL数据同步到Big Table中，同时将生产请求和非生产请求进行隔离，非生产请求直接访问ES、Cassandra，并直接通过ShardKey访问生产库，将请求隔离后，系统性能提升效果显著。

经过几年的技术沉淀，配送系统架构逐渐形成，如图5-32所示。现有系统架构实现了生产流程与非生产流程分离，非生产流程要用到的生产数据可以通过Binlog+JMQ异步实现，生产流程更加独立，可靠性、业务稳定性大大增强。



图5-32 配送系统架构

◆ 技术业务创新

①多场景众包打造配送新模式，根据不同的业务场景，众包资源池的人员进行灵活分配，通过众包系统完成派单、抢单、结算三大操作环节，为用户提供更加便利的配送服务，也为京东节约一定的人员成本。

②电子签收实现了物流全流程的电子签收，与客户交接的所有环节均可通过线上化和无纸化完成。青龙系统配送电子签收全国运营后，将带来节约运营成本、提升用户体验、让绿色物流落地三大价值。

③扫码付不仅为用户增加了一种支付体验，更降低了配送人员携带大量现金的风险。对于配送员和用户来说，移动支付避免了现金找零等多种尴尬，同时缩短了整个付款时间，提高了支付环境的灵活性。

④路径规划整合、分析老配送员的轨迹，并形成路径推荐和轨迹学习，指导新的配送员快速上手，提高配送效率。根据大数据分析和腾讯地图分析，将配送剩余时间和剩余距离展现给客户，打造更优的客户体验。

⑤通过北斗手环，实现对车辆、配送车的全程监控，并记录里程统计作为油补参考。北斗设备与智能锁车设备相结合，更完美地保护了终端载具以及货物。通过北斗设备与一体机的交互，实现报警信息及时推送给配送员。

4. 配送全覆盖——秀外慧中的无人机

快递业务一直被视作继植保无人机之后，无人机应用向行业领域拓展的重要方向之一。

事实上，目前相当一部分准备开展无人机物流业务的公司，对无人机物流的理解还停留在带着个别的货物送到消费者手中，或者消费者家的院子里。这个场景只是无人机物流体系最末端的一个环节，要知道无人机的续航半径是有限制的，这种给消费者送快递的小型无人机覆盖半径只有几公里，那么货物是如何在配送站装机的？又是如何从仓库发配到配送站的？如果消费者需要的货物附近的仓库正好没有，又是如何从其他区域的仓库调配到附近的仓库的？如果这些个环节依然使用传统的陆运方案，那么无人机物流的理念只是仅仅解决了最后一公里的问题，并不能形成一个智慧物流体系，无法给物流公司带来成本的节省和效率的提升，反馈到消费者身上的体验也就非常有限了。长此以往，除了个别交通环境恶劣的地区，其他地区的无人机配送业务将会成为鸡肋，无人机物流业务也就无法有价值地拓展起来。

为了解决无人机物流业务更深层次的问题，让通航物流网络真正行之有效地运转起来，京东提出了一个“三级同行物流体系”的概念。该网络包含了干线、支线、终端三级物流网络，其结构是：

干线无人机配送（第一级）：通过干线无人机，实现载重50~60t，覆盖1000km半径的中心仓到分中心仓的干线物流快速调拨。目前，大型客机在机场支持的条件下，是完全可以实现无人起降和飞行的，从技术上来看，干线无人机是完全成熟且可以实现的，当前最主要的问题还是在于政策监管的风险。

支线无人机配送（第二级）：支线配送是分中心之间和分中心到场站的小批量快速转运，例如生鲜、3C、高货值物品可以通过支线无人机来快速转运，这样做的好处是不必等候运输车辆，途中的损耗较小。对于这一级别的无人机，京东规划为载重200kg至2t，覆盖半径大约在500km。

终端无人机配送（第三级）：终端主要解决的是偏远地区和道路交通不便情况下的最后一公里难题，特别是偏远山区。京东无人机配送到京东在多数行政村设置的乡村推广员，这些推广员是京东在当地的重要合作伙伴，通过将无人机降落在推广员指定的场地，然后由乡村推广员再派送给最终客户。这一级别的无人机电载重量为10~50kg，覆盖半径10~50km，缺点是无法配送大件。



未来随着无人机的干线、支线和终端的联合运营，将形成一张多级联运、层层转运、天地一体、高效互通的空中物流网络，这将大大提高地面仓库的流转效率和覆盖范围。未来借助航空物流的规模化运营，将会大大减少地面仓储设施的数量，大大降低仓储成本，同时能够在控制成本的前提下，大大提升货物的流通效率。消费者在网购体验时，无论在价格上还是配送速度上，都能有大幅度提升。

在三级通航物流体系下，传统的卡车司机、快递小哥就会失业了吗？当然不是，未来的智慧物流体系将会诞生很多新的职业，这里只讲一个和无人机相关的新职业——无人机飞服师。

当卡车司机、快递小哥们升级成飞服师以后，日常的工作也将从披星戴月、风餐露宿、日晒雨淋等艰苦的状态中解脱，大部分时间只需要在监控中心负责操作无人机的运营，以及在仓库里保养、维护无人机，仅在一些特殊任务时才到现场做一些技术保障，无论是人身安全还是职业舒适度都有很大的提升。当然，这也是激励现有从业者经过学习以后升级到新职业的一种动力。

2017年7月26日，京东在宿迁的飞行服务中心正式成立，第一批50余名学员已经开始进入到正式的授课阶段，这也是“无人机飞服师”这个新职业名称第一次出现在公众视野中。从飞行服务中心培训毕业的学生，将会获得飞服师的称号，且均可以拿到民航局授权AOPA颁发的民用无人驾驶航空器系统驾驶员合格证。

无人机快递尚处于起步阶段，无论在政策层面还是技术层面，目前还面临着许多问题，我们应正视这些问题，同时也要坚信，无人机物流是未来物流的发展趋势。

5.2.4 无人机、无人车和无人仓

1. 无人机——从常态化到智能化

无人机快递业务一直被视作无人机应用向行业领域拓展的重要方向之一。作为京东智慧物流体系的一大板块，京东无人机部门在已经实现常态化运营的同时，也在不断地向智能化的方向探索和迈进，如图5-33所示。



图5-33 无人机技术智能化方向

无人机在智能化方向的进程主要包括无人机安全保障的智能化、执行任务的智能化、生产维修的智能化以及服务大众的智能化。

安全保障的智能化。在无人机安全保障智能化推进过程中，主要的技术突破方向有电子静默下的自保定为技术和智能避障技术。

◆ **电子静默下的自保定为。**主要着力推进的是电子静默攻击源追踪技术，该技术又可称为主动式反无人机技术，在军事上有着较为广泛的应用。该技术应用于无人机物流领域，可以把通航物流的安全保护从被动规避升级到主动追踪肇事者，在军民融合的应用方向有很大的市场。

◆ **智能避障技术。**避障技术在保障飞行过程中无人机和周围物体的安全性上有着非常重要的作用，使得无人机可以在复杂的环境下依旧可以自主飞行。避障技术主要研究方向为：根据障碍轮廓实现最优路径避障、识别障碍物特征，根据障碍物分类执行不同的避障策略，根据障碍物特性设计避障动作等。

执行任务的智能化。执行任务过程中的智能化推进对于无人机物流的配送服务和运营而言非常重要，智能化方向主要包括SLAM导航技术和第一视角飞行机械手技术。

◆ **SLAM导航技术。**三维空间SLAM导航技术是通过视觉的方法对场景进行三维地图重建，进而辅助无人机在受到电子静默攻击后自主行为决策，在GPS信号不稳定的情况下精准定位，重建航线内3D地图数据地理数据信息等。此项目主要采用RGBD（双目+毫米波雷达或单目+激光雷达）增量地图的实时重建，进而辅助机体在三维空间中的自主定位和自主路线规划。

◆ **第一视角飞行机械手技术。**此技术主要依赖于一个前置技术，是飞行机器人的飞控技术。此项技术基于MR眼镜设备实现手势识别以及高空作业现场影像投射，可将远程真实或虚拟的指导操作实时投射在作业现场实景中，可以代替人工执行危险性高的高空作业，在人力无法到达的区域执行细节操作，执行迅捷的现场救援或修复等。

生产维修的智能化。无人机的生产维修智能化主要包括VR装配维修和可预见性维护两部分。

◆ **VR装配维修。**VR装配维修即混合现实装配维修技术，此项技术结合VR眼镜和现有硬件形成虚拟现实环境，远程指导装配维修步骤，进而辅助训练装配维修，减少训练成本、提高训练或实操效率。

◆ **基于深度学习的可预见性维护技术。**此项技术通过各种传感器监视无人机各部件状况，收集飞行期间的传感器数据，制作一个可以预测部件剩余寿命的模型，确保每架有可能出现故障的飞机都能在出现故障之前获得及时有效的修理和维护，从而实现对无人机整体和各部件的可预见性维护。

服务大众的智能化。无人机不仅在物流行业有着广泛的应用前景，在跨行业的空中机器人平台也有着广阔的应用空间。

◆ **蜂群技术。**蜂群技术在高密度的运营航线协同规划、运营机互检、表演展示和军民融合项目中都有着广泛的应用。此技术主要包括多机间的自主感知技术、多机间的自主交互的协调网络链路及协议，以及依据长机协同动作及僚机优先级分配等群策略、群路径的



规划技术。

◆ **自适应飞控系统**。此系统使得飞控系统可以根据无人机机体的结构变化形成自适应逻辑，以配适不同的机体结构、不同的自适应缓变飞行环境、自适应突变的飞行环境等，满足未来城市间复杂环境的配适需求。

无人机的智能化探索之路不仅包括上述的各个部分，还包括对于智能芯片、新能源结构以及各项人工智能技术的探索和应用。可以预见的是，在不久的将来，无人机不仅能在物流领域得到广泛的应用，更能在人工智能领域发挥自己特有的价值。

2. 配送无人车

无人车组织架构。京东无人车团队自组建以来，发展扩张迅速。在无人驾驶与机器人等相关领域聚集起了一大批有战斗力的工程师，形成了五个战斗小组，如图5-34所示。

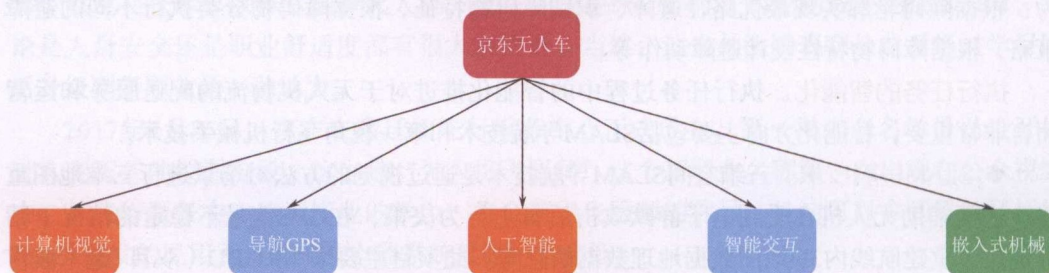


图5-34 京东无人车组织架构

◆ **计算机视觉**。SLAM技术的“控制”小组在京东商业推广化、量化的预算前提下，广泛对国内外市场的相关传感器进行算法调试与实地测试。最终，利用单目、双目摄像头及激光雷达等相关传感器完成了无人车行驶过程中的局部路径规划，可行驶区域识别及避障的工作。

◆ **导航GPS**。对于京东无人车的应用场景“最后一公里配送”来说，高精度（cm级别）的地图在全局路径规划的重要性不言而喻。而GPS定位信号对无人车运动过程中的定位十分重要，通过GPS可以得到精准的距离测量和时间戳，差分GPS技术也有助于修正误差。另外，GPS与IMU惯性导航传感器的数据融合定位方案也是另一种解决方案。

◆ **人工智能**。车辆自由行驶区域FreeSpace的识别，车道、交通灯、人等障碍物的识别与避障。通过对行驶状态的图像标注与识别，离线与在线完成相关机器学习的模型的训练，之后在嵌入式系统平台上完成部署与实时运算。最终达到识别障碍物以便安全行驶的目的。

◆ **智能交互**。语音、密码按键与无人车监控后台的交互，也包括与人的交互。京东无人车的显示屏与车上装载的智能快递货箱，在车抵达目的地后，可以与客户实时完成交互，达到人取货收货的目的。

◆ **嵌入式机械**。无人快递物流车的底层，即无人车底盘和相关传感器在国内外都是一个非成熟领域，尤其对适合于室外行驶并完成上层算法决策指令的底层响应与执行更是要求颇高。京东无人车通过对特定场景分析后，研制了新型底盘，并积极与国内相关底盘供应商合作与测试改进，研发探索多管齐下。

配送无人车的技术改进。自京东开展无人车研发以来，在研发与技术上不断探索改进，为了车辆更好地完成“最后一公里配送”的目的，对无人车的软件、硬件与算法不断进行改进。

◆ **GPS定位的探索。**GPS信号受外部客观条件的限制，如大楼中间的信号弱、雾霾雨水以及GPS接收器的高度限制等影响，在室外的行驶条件中常常会丢失差分信号，丢失定位。为了解决该问题，团队从多方面着手。第一，GPS可以使用组合导航，即IMU与GPS的组合定位，IMU可以为硬件提供运动的三维速度信息、加速度与姿态航向信息。可通过此数据与GPS信号的融合，数据标定以及补偿，达到提高精度的目的。第二，IMU可以弥补短时间GPS信号丢失的缺点，并提供高频率（100Hz以上）的数据刷新频率，提高重新锁定卫星的能力。第三，在GPS丢失信号或者无法进入差分状态得到准确定位信息时，通过激光雷达建图匹配，底层MCU运算模块根据车轮里程计和角度传感器的反馈进行运动学计算得出航迹信息，融合补偿定位，以达到弥补GPS的劣势。

◆ **无人车硬件平台的探索。**由摄像头、激光雷达Lidar、GPS等传感器融合计算得出的上层对于底层的控制。由于航向角、线速度、上下行控制/反馈通信频率的需求，底层车辆底盘的响应执行（即对控制指令如线速度、角度改变的应答以及执行）将直接决定车辆的行驶轨迹以及定位。因此，底盘的模型、电动机以及电动机驱动器的输入/输出信号需要相当的精度和可靠性。为此，京东无人车团队在汽车后驱的阿卡曼（Akerman）结构与差动四驱的底盘模型中反复测试提升效果表现。电池的动力供电也需要相当的稳定性与可靠性，为此，底层硬件不断地选型、实地测试。另外，轮速转角的反馈也需要在频率、通信可靠性、鲁棒性等方面不断优化。

3. 兰心蕙质——无人仓

继2014年成功运行的“亚洲一号”把京东的物流推向自动化之后，2016年10月，“无人仓”首度曝光。无人仓作为京东“智慧物流”技术的重要组成部分，X事业部的各类“无人仓”人工智能、机器人及自动化技术和系统在2017年618期间悉数落地并常态运营，实现了以更低的成本及更高的效率履行客户订单的目标，订单履行效率提高2~4倍。京东无人仓技术方案涵盖众多应用场景，有“地狼”机器人智能仓储系统、“天狼”Shuttle智能立体仓储系统、无人分拣智能机器人系统、分拣机器人手、自动交叉皮带分拣机和无人自动导引叉车等，为小件、中件和大件物流仓库提供不同的解决方案，为实现完整智能仓储物流链布局奠定了基础。

智能无处不在

◆ **智能算法。**无人仓一方面是用机器人代替人的腿、手、眼的功能，另一方面是用大数据和人工智能辅助来代替人的大脑进行决策。这些涉及大量的智能算法，如代替人手的六轴机器人的抓取算法，代替人眼的视觉识别算法等。

这里简单介绍一下路径规划算法思路：在批次拣货、路径规划上进行智能化决策，需要根据订单特征、工作站在途库存、订单行SKU的库存分布、AGV搬运小车的分布，统一规划拣货批次和拣货路径，如图5-35所示。

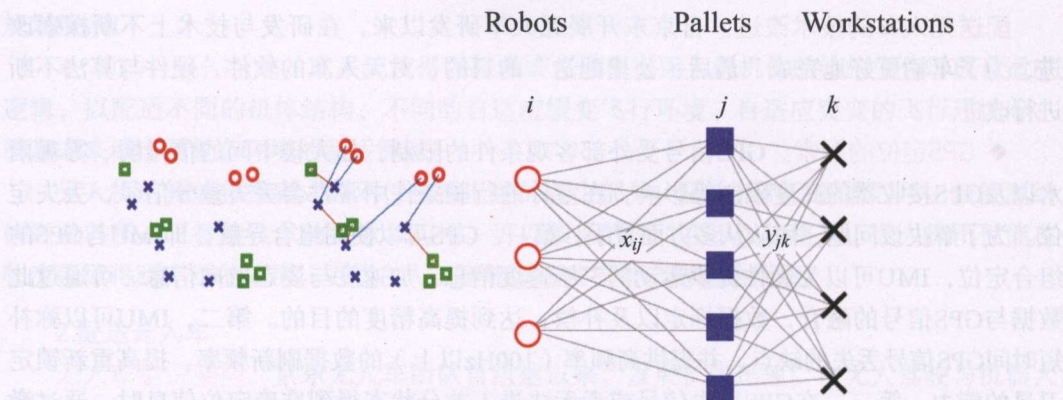


图5-35 路径规划算法

在图5-35中，○代表AGV搬运小车，■代表放商品库存的容器，×代表拣货工作站。目标是求解AGV搬运小车取到容器后，背着容器到工作站的总成本最低的指派关系，这是运输指派问题中比较难的三分图匹配。团队最初按照传统的三分图匹配的解法求解，结果运算规模过大，难以满足系统响应时间的要求。为此大家思考各种解决方式，还从美国请来了运筹优化方面的专家进行了交流沟通，在大家的头脑风暴下，发现了三分图匹配符合整数规划的特殊性，可以利用这个特征求解，终于在满足系统响应时间的要求下得到了最优解。

这中间有许多的问题要克服，如搬运机器人的转弯和直行的成本是不一样的，怎样解决呢？又比如怎样规避AGV搬运小车拥堵，让小车合理地分散到不同的路径？研发团队和美国专家合作，经过反复论证，最后通过成本修正地图和时间切片地图的方式让上述问题得以解决。

上面的指派关系和路径规划的合理性会提升出库效率，但不是全部，更重要的是在补货入库时SKU（仓库持有的品类）布局上进行智能化决策。这个智能化是为提升出库效率考虑的，包括SKU分散到的货架数，需求相关性强的SKU放在相同货架上，不同热度的商品对于工作站的分布等。这个布局的合理性对于出库效率提升意义重大，经过大量头脑风暴，我们找到了应用谱聚类的思路，求得合理的商品布局的解决方案，如图5-36所示。

图5-36说明，SKU之间的线段代表了两个SKU之间的需求相关性，强相关则线段长，弱相关则线段短。我们需要把SKU进行聚类，如封闭曲线所示，包在一起的是一个聚类，通过合理的聚类让聚类内部强相关，聚类间弱相关。

◆ **智能感知**。上面提到的是智能的部分，另外还有许多渗透到不同流程节点的智能优化。比如发现有些小车虽然电量不满但是还能工作一段时间就去充电了，非常忙时导致现场小车不够用，于是我们引入智能充电，根据任务忙闲程度和小车的电量动态进行充电决策。关于任务跟踪，开始系统是按照整个任务的状态来跟踪的，如一个出库任务完成意味着小车接到出库指令后到达储位把货架驮起送达工作站。但只监控一个大的出库任务太过粗略，异常处理也不及时，我们及时调整，开放了任务执行智能跟踪，把一个出库任务的多个步骤都进行监控，如果某个步骤执行超时提前进行预警，防患未然。还有智能导航、

对于地图拥堵的自动躲避等，还有许多这样的智能优化，无处不在，最终形成了一个智能感知的网，而且我们在不断完善这张网，让整个无人仓被智能驱动。

系统无微不至

◆ 微系统，微服务。京东始终提倡SOA化的架构，大系统小做，按职责把系统合理拆分，这同样是无人仓遵循的设计理念，而且这一过程是随着我们对

无人仓认识的深入而不断完善的。以地狼无人仓系统为例，最初把对AGV搬运小车的指挥控制、AGV搬运小车对于地图的路径资源的占用和释放放在一起，但在后面的设计迭代中我们意识到，把地图的路径资源的占用和释放独立为交通指挥系统在职责上会更清晰。它可以专门负责管理地图的路径资源，接收其他对于地图的路径资源占用和释放的请求，而对于AGV搬运小车的指挥控制则打包为小车控制系统，职责是指挥小车执行搬运任务，以及向交通指挥系统请求占用和释放路径资源。当小车数量多时，只需增加部署小车控制系统实例，交通指挥系统不受影响，如图5-37所示。

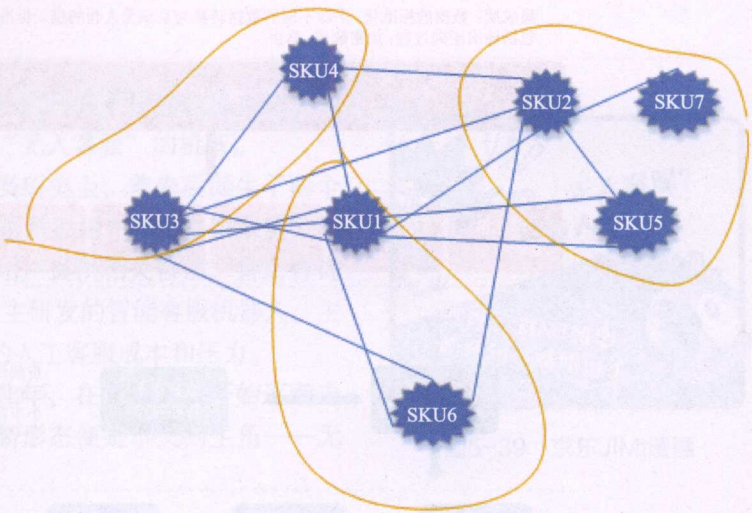


图5-36 商品布局的解决方案



图5-37 地狼无人仓系统

另外，工作站也从出入库逻辑中拆分出来，负责执行拣货、补货等任务，并且与小车控制系统交互，请求小车资源。分开后的子系统各自职责更单一内聚，系统间更低耦合，各自独立部署自主运行，每个部分可以灵活地伸缩，如图5-38所示。

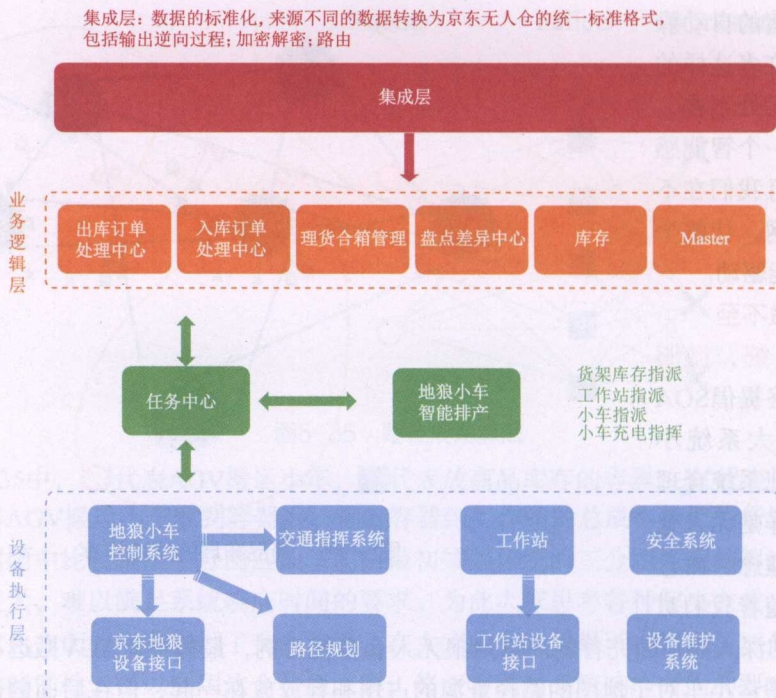


图5-38 地狼无人仓系统微服务架构

地狼无人仓系统的业务逻辑层和设备执行层分别负责处理业务逻辑和任务执行；任务中心和地狼小车智能排产负责业务逻辑层的需求，排产出可以执行的任务交给设备执行层；集成层负责数据的标准化，将来源不同的数据转换为京东无人仓的统一标准格式，加密解密和路由。

◆ **数据整合。**系统完美拆分后，各个子系统各司其职，协调工作。但是有个场景让我们感觉非常不便：智能排产系统作为无人仓的大脑，当需要查询不同的子系统数据进行决策时，最初的方案是从不同的子系统查询数据，但是这样做会导致一些问题，排产系统几乎要去查询大部分的子系统，查询的接口过多，尤其是有时结果集太大。

经过分析我们发现，这些查询是为了排产决策，不要求严格的一致性，数据延迟点问题不大，所以采用了同步数据的方案，就是把排产需要的不同服务的数据同步到一个排产决策库中，让排产服务直接查这个库，这样既提升了性能，又降低了对各个服务的压力。

“无人仓”系统和架构的设计目标是向社会提供能快速部署、可扩展、可定制、可共享的全开放式平台和服务。同时无人仓又是“无人胜有人的”智能仓库，京东会应用大数据和深度学习等人工智能技术，让无人仓为京东、为全社会的仓储物流做出更大的贡献。



Eason：物流研发部架构师，负责青龙研发技术架构，包括终端配送业务、便民自提业务、众包业务、青龙运单等业务。

5.3 智能客服

你一开口，她就懂你。这说的不是你和男（女）朋友之间的默契，而是京东“无人客服”的机智。

在京东客户服务的发展历史上，曾先后诞生了两个产品：咚咚和 JIMI。咚咚是京东用户与服务卖家进行沟通的工具与渠道，为京东用户提供相关售前、售后咨询与服务。而 JIMI 是京东自主研发的智能客服机器人，主要应对业务不断拓展带来的人工客服成本和压力。

两个产品独立发展了几年，在2017年，开始逐渐走向融合，而它们融合后的新形态便是本文的主角——无人客服，如图5-39所示。



图5-39 京东JIMI漫画

5.3.1 咚咚和JIMI历史演进

两个独立产品的发展演进路线，是如何汇合在一起的呢？下面先来看看它们各自的演进路线图。

咚咚一路走来，从1.0到5.0共经历了5个阶段。从诞生之初的草根逐步走向专业，从弱小走向规模，从分散走向统一，从杂乱走向规范，建成了今天全面微服务化、平台化与组件化的客服平台。目前咚咚正在经历5.0的融合变革之路，向新一代的智能客服平台的目标迈进，如图5-40所示。



图5-40 咚咚发展路线

京东一直致力于用技术驱动业务成长，全面提高用户体验。基于对未来客服人力成本提升的判断，以及人工智能技术的发展趋势，早在2012年，京东就决定研制智能机器人。从 JIMI 的发展路线图（图5-41）来看，JIMI的节奏更快，已经在各条业务线中攻城拔寨。



图5-41 JIMI发展路线

2012年JIMI顺势诞生，初期以售后服务为主，2014年5月开放售前服务，逐渐拓展到移动端、微博、微信等多平台端口。同时，京东也将智能机器人拓展到各个业务层面，店铺JIMI、京东金融JIMI、京东到家JIMI相继诞生。此外，还将JIMI的服务能力平台化，推出了JIMI开放平台，接入三星、长虹、华西等外部企业。

5年过去了，咚咚和JIMI两条线终于相交，基于公司未来的发展战略，提出了新的人机融合的产品形态——无人客服，如图5-42所示。



图5-42 无人客服漫画

5.3.2 产品形态与业务模式

无人客服是将咚咚和JIMI深度融合的产品，产品形态与业务模式的变化如图5-43所示。

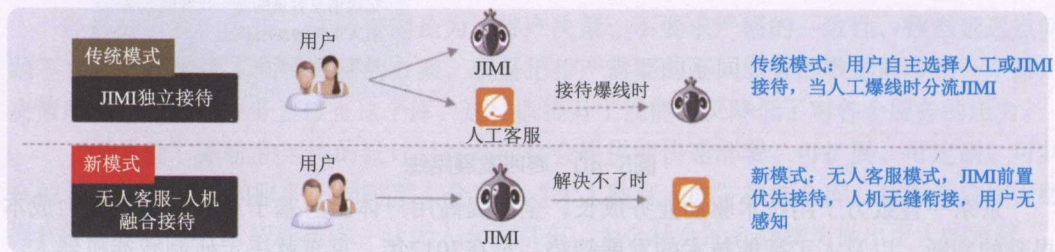


图5-43 无人客服的产品形态与业务模式的变化

从传统模式来讲，人工客服和JIMI是两条独立运行的路线，用户可以自主选择人工或者JIMI进行接待，而当人工繁忙的时候，会暂时将一部分用户分流到JIMI。而新一代的模式则是将人机融合在一起，当用户进行咨询时，首先由智能机器人JIMI来接待用户，当JIMI的回答无法满足用户的需求，解决不了用户的问题时，则无缝衔接到人工客服，用户

不会感知到整个过程。也就是说，对于用户来讲，并不会觉察到机器人和人工的分别和差异，整体的咨询流程十分流畅，大大提升了用户体验。同时，这种模式也缓解了人工客服的压力，提高了客服的工作效率，节约了大量的人力成本，如图5-44所示。

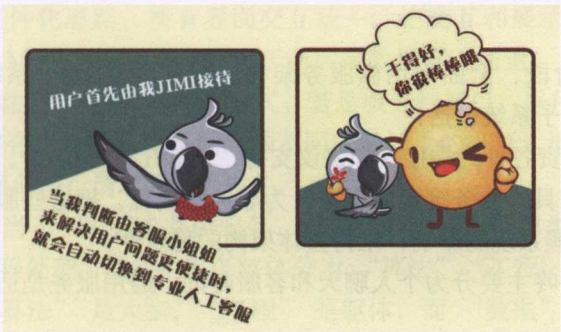


图5-44 无人客服人机融合接待

5.3.3 京东咚咚和 JIMI架构体系

1.融合架构体系

产品形态和业务模式的巨变，对两个产品各自发展了5年的技术架构体系也带来了巨大的挑战。从技术层面上看，咚咚在通信方面更有优势。所以，从整体架构上，咚咚统一对外提供通信层的接入。而 JIMI 作为后端智能服务，则通过服务网关接口进行异步化桥接，如图5-45所示。

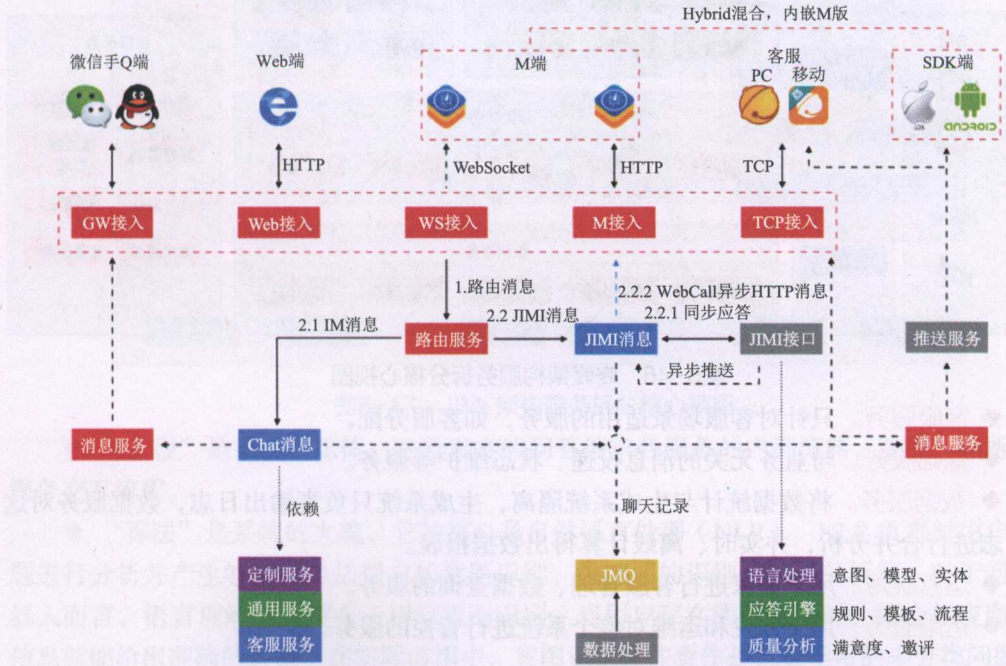


图5-45 无人客服融合架构



在架构融合的基础上，咚咚接管了原本 JIMI 需要服务的外部终端和通信入口，而 JIMI 则专注做好后端智能服务，让专业的技术团队专注做好各自专业的事情，提升效率并降低成本。而在它们各自独立的技术架构领域，又各具特点。

2. 咚咚架构体系

咚咚在4.0破茧阶段，全面采用微服务架构，拆分出了8大子系统，50多个核心服务（图5-46）。其中8大子系统包括：

- ◆ **接入服务**。负责各种终端接入和协议支持，比如HTTP、TCP、WebSocket 等。
- ◆ **定制服务**。和具体业务绑定的服务，为非标准流程，把这些服务独立出来，可以避免核心架构受外部依赖系统的影响，维持整体架构的稳定性。
- ◆ **通用服务**。咚咚主要分为个人聊天和客服咨询，通用服务是适用于这两种业务场景的服务。



图5-46 咚咚架构服务拆分核心视图

- ◆ **客服服务**。只针对客服场景适用的服务，如客服分配。
- ◆ **基础服务**。与业务无关的消息投递、状态维护等服务。
- ◆ **数据服务**。将数据统计与生成系统隔离，生成系统只负责输出日志，数据服务对这些日志进行合并分析，并实时、离线计算得出数据报表。
- ◆ **运营服务**。用于商家进行客服管理、数据查询的服务。
- ◆ **运维服务**。用于开发和运维对整个系统进行管控的服务。

3.JIMI架构体系

如图5-47所示，顶部展示的是目前 JIMI 支撑的所有业务场景，包括：京东自营业务的导购、售后和虚拟业务，京东 POP 店铺机器人，京东金融机器人等。下面是统一接入层，界面按照终端接入组件化思路，所有界面交互统一，把交互和展示逻辑进行封装，按照 Web（PC）、SDK、M 页面分为三类。对于前后端数据交互通道，浏览器采用HTTPS 和 WebSocket 方式，移动端 SDK采用TCP 方式，若是服务端接口，采用京东内部标准化 JSF RPC 方式。这样就整合了所有来自不同终端不同业务场景的机器人请求，所有请求统一转发到“调度引擎”服务进行请求调度分发。

“调度引擎”相当于一个请求路由服务，根据终端机器人请求类型的不同，会调度分发给后端不同服务流程处理。后端的服务整体分为“算法”“工程”“数据”三类。对于聊天机器人来说，“算法”是大脑，“工程”是躯体，而“数据”则是血液。下面分别简单介绍三类服务的作用与协作方式。

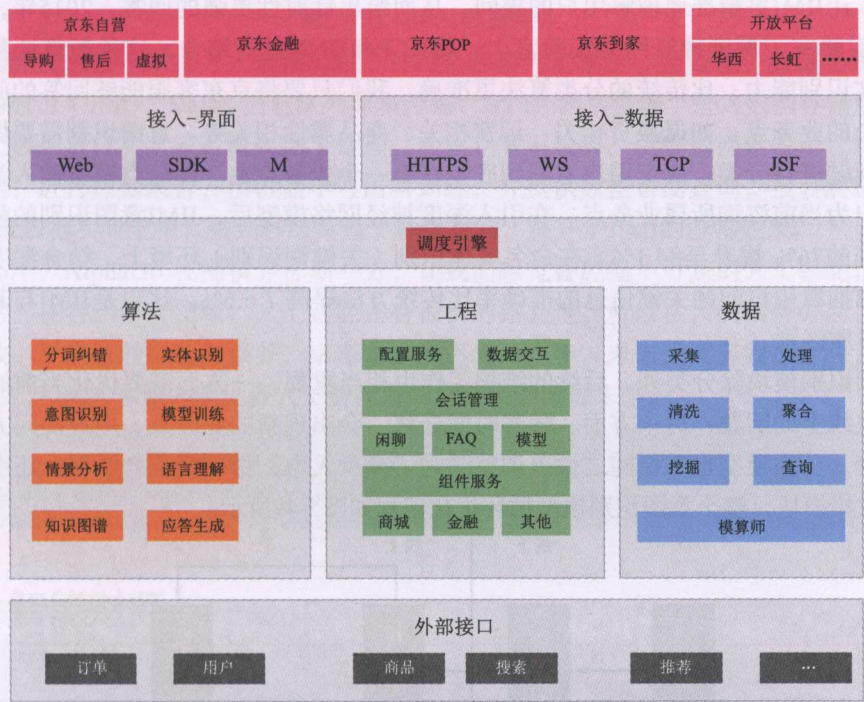


图5-47 JIMI 架构服务拆分核心视图

- ◆ “工程”是系统的躯体，它负责在线问答请求各服务的内部流转，如图5-48所示的服务交互流程。
- ◆ “算法”是系统的大脑，它的核心是自然语言处理（NLP）。NLP 负责对用户的问题进行分析并产生答案，包括用户的意图识别、关键词的识别、歧义分析等。对于问答机器人而言，语言理解的关键在于用户意图识别。当用户存在明确意图时，结合相应商品等信息就能给出准确的答案。在实际应用中，意图识别往往看作是机器学习的多分类问题。

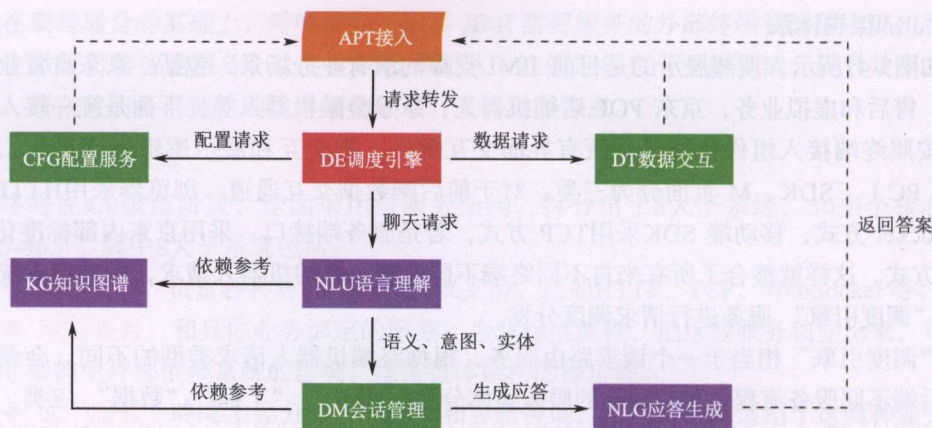


图5-48 JIMI 服务交互视图

为了让JIMI更精准地理解用户的提问，从而给出针对性更强的回答。2015年，京东成立DNN实验室（深度神经网络实验室），将NLP和DNN进行结合，这种新算法具有一定的上下文识别能力，比传统的分类算法更准确。我们只要将京东客服能够回答的问题归纳成一个个的业务点：如退换货相关、运费相关、商品属性相关等。意图识别需要做的就是对用户问题进行分类，决定用户意图到底是在哪一个业务点上。即分类模型输入为用户问题，输出为当前咨询所属业务点。在引入深度神经网络模型后，JIMI意图识别的整体准确率由原先的76% 提升至84.1%。在命名实体识别（关键词识别）环境上，结合深度神经网络，JIMI抽取用户问题关键信息的准确率比传统方法提高了6.6%。这也是JIMI与其他同类产品的本质区别。

意图识别模块除分类外，后续的维护工作也相当重要。一方面需要优化当前的分类效果，解决线上的问题；另一方面，随着时间迁移，会出现新的业务点，用户提问方式也会发生改变，分类模型也需要随之做出调整。为了减少人力，做到模型自学习，还引入了线上数据收集模块，整个意图识别模型循环优化流程如图5-49所示。

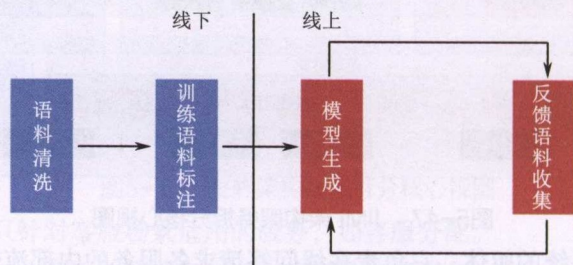


图5-49 JIMI 意图识别模型循环优化流程

这样线下和线上的工作联动运作，就能保证JIMI的意图识别不断优化，跟得上业务的不断升级。

◆ “数据” 是系统的血液，JIMI数据架构体系如图5-50所示。

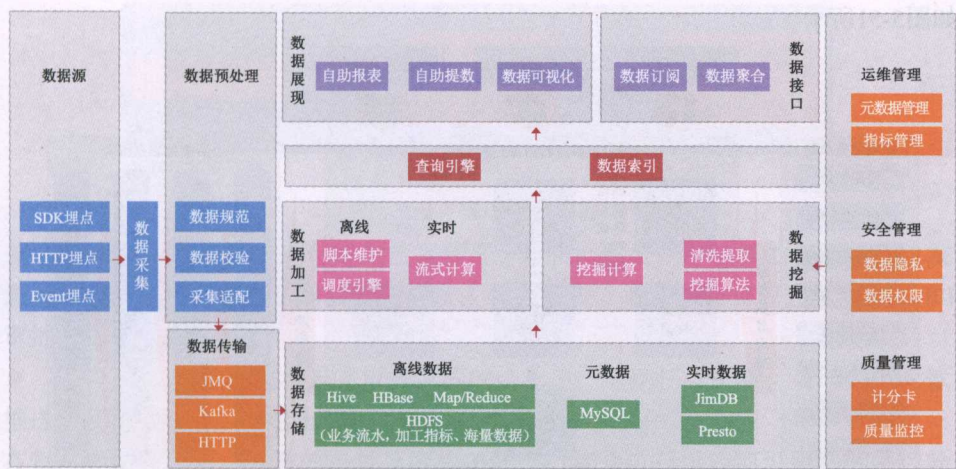


图5-50 JIMI 数据架构

数据经过采集（采集方式包括SDK、HTTP 和事件采集）、预处理、数据格式校验，通过传输通道（JMQL、Kafka 消息队列异步传输、HTTP 同步传输）进入数据存储层。通过建立的元数据进行实时或者离线加工处理，并建立索引服务，供查询搜索使用。对于清洗后的数据，进行数据聚合，供机器学习语料训练；并提供数据自助查询，报表展现。能支撑 10 亿级数据实时上报，做到秒级延迟、秒级查询，提升了JIMI的应答效果。

2016年，AlphaGo 在围棋界称霸一方，让我们感觉曾经遥不可及的人工智能正在走进现实生活。在自然语言领域，人工智能还有很长的路要走。所以在人机结合方面，为了提供更高效、更满意的沟通与服务，京东才刚刚迈出一小步，未来无人客服也会一直进化下去。让我们和无人客服一起见证人工智能发展所带来的美好未来。



mindwind：京东成都研究院资深架构师，技术委员会负责人。
虎哥闻仙：移动研发部无人客服项目负责人。

5.4 智能财务

5.4.1 结算的秘密

1. 自营供应商结算——功到自然成

做财务系统的精髓是必须不差一分一毫，既要保证数据的流转快，还要保证数据准确度高，更要保证整个系统的稳定性。

自营供应商结算系统（以下简称自营结算系统）主要实现京东自营结算业务与供应商相关联的各类业务，主要包括采购结算、采购退货结算、售后退货结算、返利结算、预付款结算、质保金结算、广告结算等业务。针对前端供应商关联业务提出的各类供应商结算申请进行确认和控制，并提交给资金系统完成结算，确保供应商结算款项安全。系统交互



框架如图5-51所示。

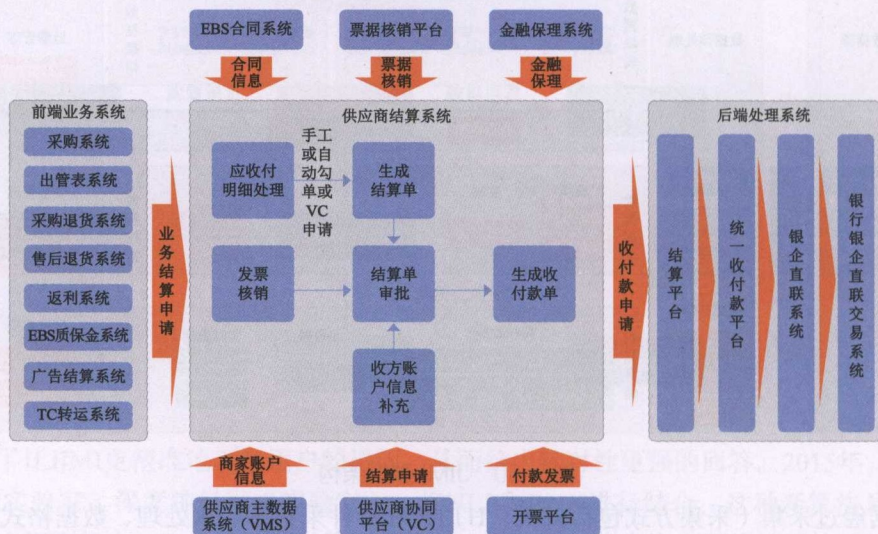


图5-51 自营结算系统交互框架图

系统按照功能模块拆分成计费模块、结算模块、审核模块、付款模块。每个不同的应用模块，可以单独部署，互不干扰，如图5-52所示。

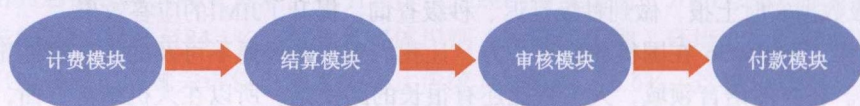


图5-52 自营结算系统模块图

◆ **计费模块。**集成库存生产的自营相关的全量数据，搭建了一套最小粒度的结算基础数据，采取T+1汇总方式，将数据发送至结算模块参与结算。自营业务一直是公司的重头戏，入库、销售、退货等明细数据，经过逻辑批次拆分到每一个采购单号，日增长量日益庞大。因此，团队采取了MySQL分库分表机制，将已结算的历史数据储存至HBase。

为保证数据准确性，系统采用“Version—版本号”做严格的库存控制，技术上最直接的问题就是防并发。如果销售数据的总数量超过了当前采购入库的总数量，则该笔销售数据将被列入“异常名单”。

◆ **结算模块。**作为结算系统的核心功能，结算模块由最初的手工提交申请，扩展为系统自动勾单，为业务人员减轻了50%的工作量。自动勾单是自营结算系统的一个特色功能，通过集成采购、退货、返利及其他应收等系统数据，系统按合同账期及风控值来设置勾单逻辑，系统自动生成结算单。任务驱动机制实现每天运行一次自动勾单，将配置表同步到Redis缓存中，利用MQ分发机制并行处理。财务业务人员通过配置自动勾单配置表，实现不同商家按照不同的账期类型进行结算，严格控制倒挂风险，自动勾单原理如图5-53所示。

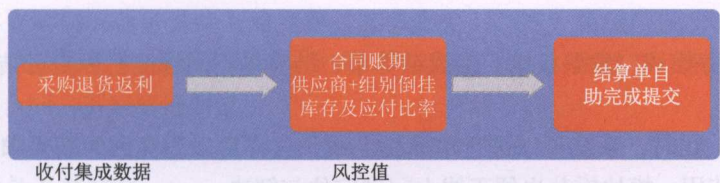


图5-53 自动勾单原理

◆ **审核模块。**对结算单进行审批流程管理，审批时需要校验结算的合同信息、发票核销信息及供应商账户信息等。对接公司的统一代办流程，业务人员可以通过手机端和PC端随时审批。

◆ **付款模块。**付款模块将走完审批流程的结算单，利用任务机制，自动创建付款单，自动推送至收付款平台，交给出纳相关部门进行付款。

在每一个结算环节、上游的退货系统建单环节，都会进行供应商可用余额的校验，防止新生成的应收数据造成商家倒挂。利用大数据平台，实时计算汇总金额，为公司的资金管理筑起一道防火墙。

随着公司业务的迅速发展，自营结算系统作为末端服务系统，能及时有效地支撑每一个新开展的业务场景。自2012年开始，结算明细类型从采购入库单类型、采购退货单类型、返利类型、质保金类型、售后退货单类型、其他扣款类型6大类已经扩展到23大类。同时增加了许多新的业务模式，如正向销售数据、逆向销售数据、仓储服务费、TC运费、公益款、广告费等，对于扩展的一个新业务类型，系统只需要很短的时间就可以完成新业务的部署上线。

2. POP商家结算——破竹之势

POP T+1商家结算系统主要负责将上游业务系统统计的商家入驻、使用京东开放平台过程中产生的各种费用明细，在协定的业务时间点T加1的时间内转换成收付款单，发给下游网银钱包，完成收付款结算。

T+1是指触发结算的日期T然后加1天，T在不同的商家与京东合作模式和不同的结算业务中选取的时间点不同，不论T选取哪个时间点，都会在T基础上延后一天完成整个结算。POP商家不同合作模式或业务中T含义如图5-54所示。

整个系统按照几大模块进行拆分，并且单个模块功能节点实现可配置化：

- ◆ **接口模块，**为其他系统提供接口服务和消费其他系统提供的接口。
- ◆ **消息模块，**消费其他系统传递的消息。
- ◆ **任务模块，**单据打标是否结算。
- ◆ **Web服务模块，**为业务方提供

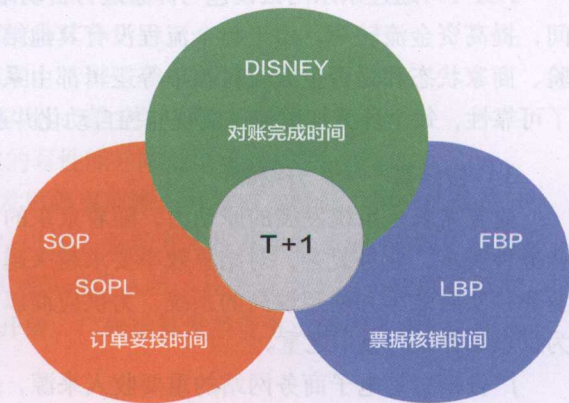


图5-54 POP商家不同合作模式或业务中T含义图



Web页面服务。

◆ 内部业务模块，发票核销、生成结算单、结算单不同维度打标是否结算和生成收付款单。

在新功能上线时，系统可以进行单模块上线，这样可以避免因为新需求上线异常影响其他功能模块使用，模块拆分也便于线上问题定位与解决。

POP T+1从接收明细到生成结算单，然后汇总结算单到最后接收网银钱包收付款成功消息，整个过程无须结算人员控制。为实现不同业务和相应的财务规则有效匹配，团队引入了锁的概念。如商家未进行发票核销的欠票锁，商家刷单导致金额异常的大金额锁，商家未收款大于未付款的应收未收锁，还有公司锁、手工锁等。将不同业务生成收付款单的过程划分成一个一个节点，每个节点是该业务实际需要验证的锁。当需要增加或减少锁时，只需修改节点配置即可。同样，当有新结算业务时，会和原有业务锁匹配，这样新结算业务只需要配置即可完成整个需求开发。

POP T+1技术在实现上采用了数据库分库分表、读写分离、GUAVA缓存技术和HBase数据迁移历史等技术。

POP开放平台目前有14万多家店铺，每天对应的明细数据量为1700万条左右。对于大量的明细，团队首先按周进行了分库处理，单周在A库，双周在B库，其次按照商家编号进行哈希分表。

接收明细接口需根据上游的费用数值获取数据库内储存的费用名称和收付方向，采用公司封装的JIMDB分布式缓存技术，对费用进行缓存。在一定程度上可以减少对数据库磁盘的读操作，提高数据库性能和接口性能。

生成结算单规则预处理，为提高生成结算单汇总效率，预处理拼接生成结算单的财务规则，生成一个结算单业务字段，后期按单列汇总即可。如原生成结算规则为明细中商家编号、费用编号、妥投时间和模式的四列集合操作生成一个结算单，现需商家编号、费用编号、妥投时间和模式拼接的一列汇总生成一个结算单。

任务并发生成结算单按明细的商家编号进行哈希分表设计，同一个商家的费用明细在同一个表中，而生成结算单规则中包含商家编号，从而实现并发汇总明细生成结算单。

POP T+1通过京东网银钱包与商家进行自动结算，避免人工转账，大大缩短商家结算时间，提高资金流转率。由于整个流程没有其他结算系统人工审核过程，因此结算单金额校验、商家状态和是否生成收付款单等逻辑都由系统自动执行，从而节省了人工成本，提高了可靠性，每个环节加监控，实现监控自动化并避免结算风险。

3.广告结算——广开财路

业务需求是系统发展的原动力，随着京东的飞速发展，大量的业务接入到结算业务，业务种类多，业务复杂，对结算效率要求越来越高，那么建立一个可扩展性强，兼容多种业务，实时效率和稳定性高的系统，为供应商以及广大的消费者提供满意的结算体验，成为京东结算人的重中之重。

广告销售是电子商务网站的重要收入来源，过去没有对广告财务数据进行线上管理，造成大量手工作业和数据统计困难。因此需要建立广告销售的财务系统，即广告结算。广

告结算目前主要面向的业务有商务舱业务（RTB业务）、按天展现广告（CPD业务）及京腾计划。商务舱业务是实时投放的广告模式，例如京东快车、京挑客、京东直投。CPD业务是按天投放合同，例如京选展位。

基于上述的业务主线，广告结算系统架构如图5-55所示。

由广告业务系统传送基本商家签订合同信息，根据合同信息建立以合同及商家维度的各项费用的账簿。财务人员在广告结算系统做相关审批，系统自动记录各项金额变动情况，并把收付款单传送到收付款平台，在发票平台进行开票，从而让广告的结果业务线上化。

合同信息。合同信息中的合同编号和甲方编号是广告结算的结算维度。在创建合同的同时，创建了合同与甲方维度的账簿，用来记录各项类型的费用，例如广告费、押金、预存款等。

费用账簿的费用采用垂直分表技术，费用账簿按业务分成3张表，3张表有相同的合同编号与甲方编号，分别记录收退款费用、开票和投放费用以及业务间的转款费用。

收退款信息。收退款信息采用收款退款同表以类型区分的方式。起初收款是分别独立的收款线路，新业务接入不能统筹兼顾，甚至有防重冲突，特别是后续全网DSP及京腾计划的接入，需另外开发，团队对此情况进行了如下改造：

- ◆ 加系统来源，不同业务以系统来源清晰区分。
- ◆ 防重采用统一枚举前缀，直观易识别。
- ◆ 采用算法封装设计模式，提炼收款共性，建立统一算法模块。
- ◆ 数据库采用水平分表，提高收退款的写性能并降低了单库的数据量。

经上述改造，业务接入速度加快，开发思路更清晰，开发成本降低，用户体验提升。

任务驱动及MQ交互方式。广告结算是以任务驱动方式将开票状态MQ送到广告业务。若开票状态有错误，MQ将重试以报警方式通知开发人员，任务驱动支持失败重置重发，提高系统可操作性与容错性，实现实时快速开票。广告结算发票业务流转图如图5-56所示。

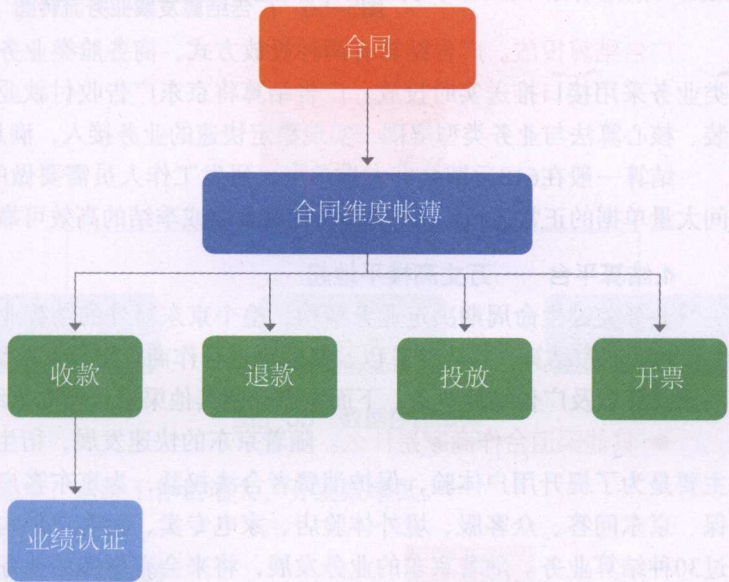


图5-55 广告结算系统架构



图5-56 广告结算发票业务流转图

广告结算投放。广告结算有两种投放方式，商务舱类业务按照轮询Worker投放，CPD类业务采用接口推送实时投放。广告结算将京东广告收付款业务线上化，通过核心算法封装、核心算法与业务类型解耦，实现稳定快速的业务接入，满足投放广告商的需求。

结算一般在618后期会有大量单子，研发工作人员需要做的还有很多，既要保证618期间大量单据的正常运行，又要保证618后月结或季结的高效可靠。

4. 结算平台——万丈高楼平地起

业务关键生命周期决定业务架构，整个京东对外的结算业务，主要包括自营供应商结算、POP商家结算、广告结算以及其他渠道合作商家结算。前面已经介绍了自营结算、POP商家结算以及广告结算业务，下面重点介绍其他渠道合作商家结算。

◆ **其他渠道合作商家是什么。**随着京东的快速发展，衍生出很多业务条线，这些业务主要是为了提升用户体验，保护消费者合法权益，为京东客户提供更优质的服务。例如延保、京东问答、众客服、境外体验店、家电专卖、大客户采购、司法拍卖、售后维修等超过30种结算业务。随着京东的业务发展，将来会有更多的业务条线。结算面向的对象可能是一些不同规模的公司、法院、普通个人、作者、线下合作门店等。

◆ **结算平台诞生。**为了满足京东快速多变的业务发展模式，京东构建了一套具备易扩展、易维护、高复用的结算系统——结算平台。它承载了京东除自营和POP业务条线外95%以上的业务条线。结算平台是以业务结算核心生命周期来构建的，包括基础数据、应收应付账、票据管理、结算管理、台账管理、收付款单管理六大模块。它们之间相互协作，共同构建业务完整的生命周期。系统详细功能如图5-57所示。

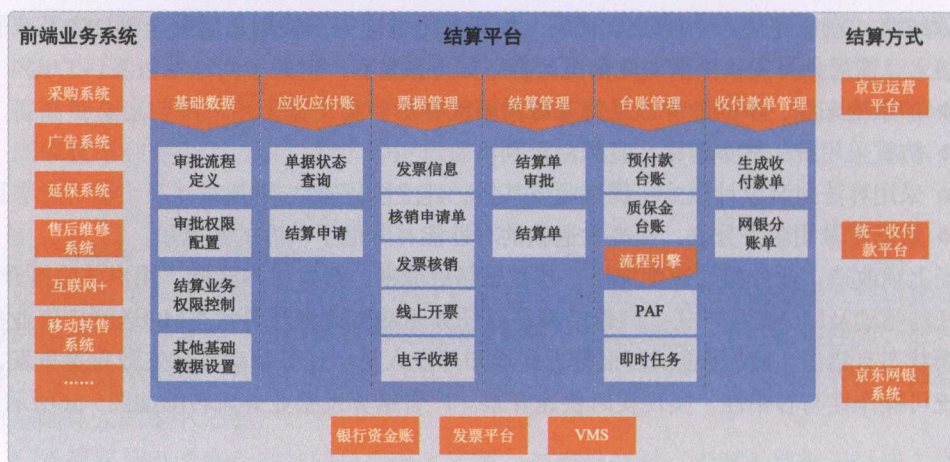


图5-57 结算平台系统详细功能图

◆ **结算平台的坑。**下面谈谈结算平台遇到的坑。结算平台是针对结算生命周期定制的平台架构，包括任务调度框架、即时任务调度插件、审批流插件、动态规则引擎等技术。再次对“应收应付账”这个关键环节做下讨论。应收应付账是与商家往来的收付款信息，涉及各种各样的业务条线。那么我们怎样储存这些结算业务数据呢？

起初，这种渠道商家业务较小，上线紧急，因此采用标准表+扩展表来储存数据，数据存储结构如图5-58所示。

在此种存储模式下，各业务页面需要自己定义，数据查询等业务操作需要重写标准接口及其他实体等。随着京东业务的迅速扩展，标准表+扩展表已远远不能满足业务需求及数据增加的速度。同时，各业务高耦合到牵一发动全身的地步，运维工作量增大，开发效率低。

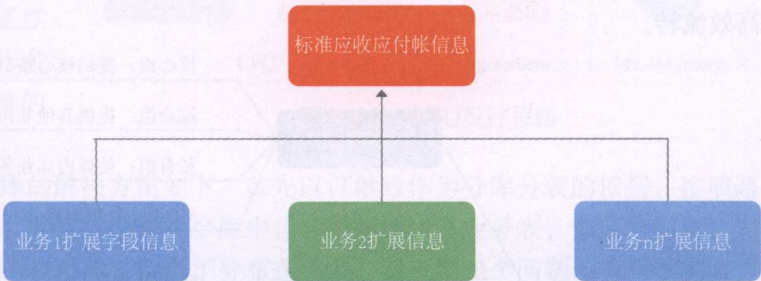


图5-58 数据存储结构

为了满足京东业务条线的快速发展，必须重写结算平台。团队提出了新的数据存储方案，数据存储结构如图5-59所示。

数据存储貌似没有变化，但仔细看就会发现不同。按照业务条线分表，可分为标准业务条线及差异业务条线，每个表都是全量的业务信息。并且同一种业务可以按照规则分多张表，通过动态列配置保证平台的易用性和可扩展性。

随着京东业务条线的不断扩张，结算平台的不断成长，业务接入更加可靠、工作效率提升数倍，同时减轻了测试部门的压力，相信结算平台会做得越来越好。

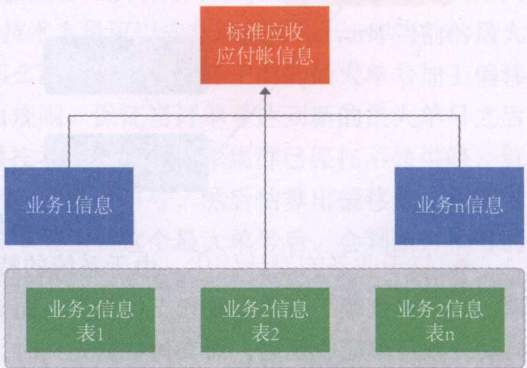


图5-59 数据存储结构

5.4.2 你下单我算账，你不知道的来龙去脉

1. 订单台账——财务数据的天平

订单台账系统是下单流程的核心系统之一；为集团各个业务部门提供准确的订单财务数据。订单台账系统就像一架天平，一端是应收账，另一端是实收账，提供精准可靠的对账服务，不偏不倚，分毫不差。



◆ **复杂业务系统拆分。**台账系统业务流程复杂，正向应收实收对账拆分，逆向冲销退款控制，同时需要提供各个维度的查询接口，着实是一个庞大的系统。台账系统在京东下单流程中处于核心地位，因此保证系统稳定性就显得十分重要。财务研发部根据业务性质的不同，把复杂的系统拆分为多个独立子系统。

如图5-60所示，台账系统虽然业务复杂，但逻辑清晰，功能划分合理。台账服务是所有子系统的核心，技术人员把台账服务进行二次拆分分组，区分核心接口与非核心接口、区分对外接口与内部处理逻辑，将最优的资源分配给最核心的业务，保证核心业务的正常高效流转。

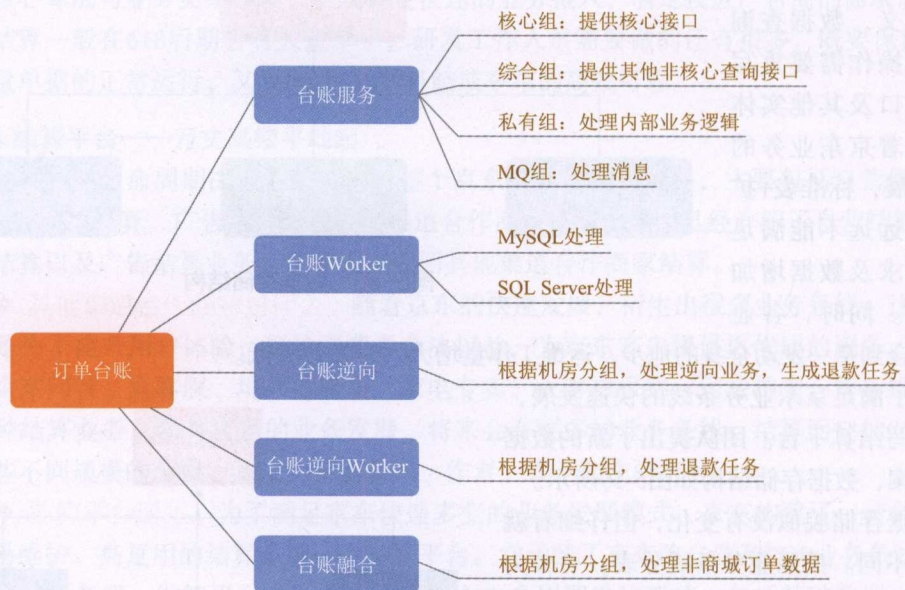


图5-60 台账系统结构图

◆ **基于业务的设计优化。**由于系统的特殊属性，台账系统并不能与其他系统完全解耦，而是强依赖于订单系统和支付系统，而支付数据依赖于订单应收数据。当网络中发生抖动导致订单应收延迟，而支付数据先到时，技术人员会直接返回支付系统失败。一旦网络出现问题，这种情况会大批量地发生，极有可能对支付系统造成难以承受的压力。

另外，当业务发展到一定量级时就会出现一些比较棘手的问题，像如何提高吞吐量？系统间接口调用的速度已经远远大于接口提供者的处理能力，如何让接口不成为整个业务线的瓶颈呢？

如图5-61所示，技术人员从CPU的设计思路上寻找到灵感，当磁盘远远跟不上CPU的读写速度时，才设计了内存。随着CPU的发展，内存的读写速度也远远跟不上CPU的读写速度，因此，为了解决这一问题，CPU厂商在每个CPU上加入了高速缓存。

技术人员需要的正是这一层缓冲，将数据先落下来再异步化并且内部消化，这样既可以提高吞吐量，又可以解决前面所说的问题。于是，技术人员将所有的核心流程全部异步化处理，压测结果显示50台Docker集群核心接口的吞吐量可以达到300W/min，而且TP99性能在70ms以内，无论是吞吐量还是性能都得到了很大的提升，可以完美应对618的挑战。

◆ **灵活的分库分表。**台账系统面临的重要挑战之一是如何解决海量数据存储的访问问题，最常用的解决方案是分库分表技术。对于此技术，目前各家公司都有不同的方案；但是都需要解决两个问题：一是如何保证在分库分表的前提下避免分布式事务，二是如何保证库的平滑切量和弹性扩容。

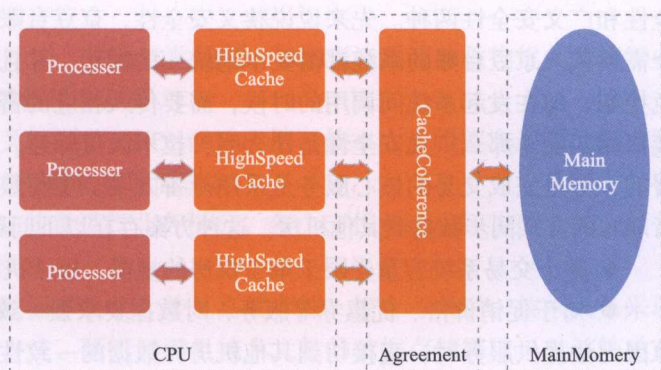


图5-61 CPU设计思路

对于这两个问题，台账的解决方案如下：首先以订单号作为分库分表的依据，按照路由规则路由到同一个库中，并且保证同一个库中包含所需的所有业务表，如此可以解决对于一个订单下的事务操作不会跨库，避免了分布式事务。对于第二个问题，有两个场景：一是数据库的切库操作，比如从SQL Server库迁移到MySQL库；二是618时数据库的弹性扩容。对于上述问题的解决方案，台账采用多段式组合分库分表方式，即将路由规则分为多类，每一类对应一种路由规则。例如，对于单号在0~100范围内的订单可以分为两段路由：0~50、50~100，相对应的路由库则为：SQL Server库集群，MySQL库集群。当技术人员需要从SQL Server库切到MySQL库的时候，技术人员可以先获取SQL Server库当前的最大单号，然后根据业务量来估算一个偏移量，那么新的最大单号等于旧的最大单号加上偏移量，就可以在不重启的情况下，动态配置路由规则，保证当订单号达到新的最大单号之后会切到MySQL库集群。而在618的时候，如果技术人员发现主库集群已经扛不住洪峰，就可以启动弹性扩容机制动态扩容，即记录下当前的最大单号，然后估算出偏移量，用旧的最大单号加上偏移量作为新的最大单号。当订单号达到这个最大单号后，会判断订单号的最后一位是奇数还是偶数，以此作为依据平均路由到主库集群和扩容库集群，让其各自承担50%的量，从而减轻主库集群的压力，保证了库的弹性扩容。

2. 小京豆大流量——京豆台账的使命

在京东商城上购物，你会发现有很多地方赠送京豆，比如签到送豆、评价送豆、购物送豆。也有很多地方可以使用京豆支付，比如购买实物商品下单使用京豆，购买虚拟商品使用京豆等。所有的交易流量都要进入京豆台账进行统一汇总，并且计算出用户在进行一系列操作后的京豆余额。所以京豆台账扮演的是一个会记账的会计角色。

京豆台账的使命。作为一个记账系统，京豆台账需要做好以下三方面工作：

- ◆ **记账要安全。**京豆就是钱，每产生一笔京豆交易京东都会产生相应的收入和支出。
- ◆ **记账要清楚。**每一笔京豆的支出和收入，均要在用户的京豆总额上得到体现。
- ◆ **系统要扛得住量。**京东商城主页上可使用京豆的地方有数百个之多，而所有的操作最终都要汇入京豆台账，所以服务器要能承受得住大流量带来的压力。

京豆台账的功能。京豆台账以一个服务端的形式存在。对于安全性，可以分为狭义安



全性和广义安全性两种。先来说说狭义安全性，京豆台账有一系列的安全校验机制，每一个需要接入京豆台账的系统都需要申请统一的密钥，并且对所申请的密钥要有相对应的角色控制，即在发起系统间调用的时候，需要传入相应的密钥，并查看密钥是否与所申请的接口相匹配。那么广义安全性是什么呢？技术人员知道，交易系统现阶段是同城多机房部署的，但是完成交易的核心服务是采用冷备方案。主写数据在一个机房，通过主从复制或者双写的方式同步数据到其他机房。这种方案存在以下问题：

- ◆ 整个交易系统容量受限于单个机房的规模，每个机房储存的是全量交易数据。

- ◆ 对于促销价格、优惠券等服务，对数据要求强一致性，当主写机房因为网络故障等原因无法提供服务时，直接切到其他机房，数据的一致性无法保证。数据写服务不是双活的。

- ◆ 同城多机房方案在面临如地震、城际骨干光缆被挖断等极端情况下，容灾能力也比较脆弱。

为了保障安全，我们采用了多中心交易实现多个异地机房同时提供交易服务，任何一个机房出问题后都能由其他机房接管。用户尽可能在最近的机房完成交易，并且在单机房内形成流量闭环（将用户数据封闭在一个机房，解决了数据复制延迟带来的后发先至的问题。实现单机房流量闭环，需要移动网关、单品页、购物车、结算页、优惠券等多个系统路由规则一致，统一合作）。

关于清晰记账，每一个接入京豆台账的系统都要有相应的活动支出费用。京豆台账系统会把活动所申请的京豆数量配置在相应的京豆活动池里，用来控制此类活动可以发起的京豆赠送数量。此类活动调用发放的每一笔京豆都需要从京豆池里扣除，这样既可以保证活动的效果，又可以通过控制京豆池来动态控制活动时间和活动支出。再就是保证商城用户支出和收入，京豆的每一笔操作都要和用户的京豆总量保持一致，这里技术人员可以用数据库的事务一致性来确保。

3. 浴火重生的POS直联系统

POS直连系统是一个连接后款订单生产和财务对账系统，它兼顾着后款订单在配送时的收款查询和支付确认，以及收完款后的财务对账两大关键流程。它是处理商城后款订单生产线上上的一个关键环节。

POS直连系统大致分为三大部分：网关、前置、ERP系统。网关主要负责与银行、网银的通信；前置主要给网关提供接口服务；ERP部分主要负责订单的支付与实收的财务对账。对账部分按支付方式分为刷卡和现金两部分，按订单类型分为自营订单和中小件外单。

随着公司业务扩张、收款方式增多及京东物流的不断开放，POS直连系统原有收款和对账方式越来越难以应付与日俱增的流量，系统维护成本越来越高，并且在存储、时效、性能等各方面也难以满足业务需求。给系统动刀势在必行。要动刀，就要找到痛点，只有找准了问题，才能一刀见效，彻底解决问题。技术人员总结了两大痛点：订单业务类型增多、收款方式增多；现金回款对账低效复杂、财务人工成本高。最后经过研发、产品、财务、配送几方的调研评估，最终决定重构。整体重构思路为：业务拆分、隔离、业务流程

细化简化、减少人工介入。

POS前置重构——简单高效可靠。针对这些问题，POS直连系统的重构优化着重于业务拆分、整合，数据拆分，流程细化几个方面。

针对在订单生产流程中处于关键环节的POS前置应用，技术人员希望在逻辑处理方面做到周密但尽量简单，数据存储方面做到可靠而高效。

带着这样的愿景，首先对POS前置进行业务拆分，各个支付渠道进行物理上的分隔，确保做到各渠道业务独立，互不影响。

解决数据存储时，优先确保数据不丢失，并且高效存储。综合考量后，技术人员采取常规的按订单号进行分库分表的方式，既可以满足技术人员的数据存储需求，又够简单高效地完成数据路由。

在POS前置进行业务量切换时，出现了一个棘手的问题：如何平滑地切换流量。最终技术人员确定以POS机终端号为维度，从一个配送站点开始切换，这样就可以把影响范围控制到最小，而且能及时联系到站点相关操作人。

推出POS掌柜——现金回款对账无忧，结果清晰直观。一直以来，后款现金订单的货款都是一个风险很高的问题，技术人员希望在尽量减轻配送、财务人员的工作量的同时，能够将货款风险降至最低。因此POS掌柜系统的着力点也就变得清晰：回款支付入口整合、实时对账、实时结算。

◆ **回款支付入口整合。**由于订单量大幅增长，后款订单这一京东特色服务也随之猛增，后款订单的货款金额增长明显。对于集团来说是极大的好事，但对于财务和配送的工作人员来说却是件令人挠头的烦心事。

一方面财务研发担心货款的回款风险；另一方面，配送站的站长们每天配送结束后，手里握着大笔的货款，如果不能当日及时回款，他们会寝食难安。所以POS掌柜的首要任务就是解决这种不良的用户体验。

在POS掌柜中，技术人员整合了原本存在于资金归集系统的回款支付入口。这样，POS掌柜就可以在整个后款订单的回款入口上着手，将整个业务流程把控起来。

用户也无须多次切换系统来确认回款情况。回款流程也重新进行了梳理，按照配送的需求，支持按照妥投订单金额进行回款，尽量减少人工操作。

现在POS掌柜的回款，只需要点几个按钮即可完成，既不需要人工计算回款金额，也不需要录入卡号等相关信息。在降低工作量的同时，也确保了精准回款，同时为实时对账、实时结算打好了基础。

◆ **实时对账，实时结算。**作为主要面向财务人员的业务系统，技术人员更清楚财务的痛点，对账和结算是财务核心的两个业务，也一直是个老大难问题。仅仅一个对账，就能让财务忙得焦头烂额。

经过认真调研，多次与业务人员沟通后，技术人员确定了回款即对账，对账即结算这两种解决方式。但是这样会将的对账、结算与回款严重耦合在一起，所以技术人员采用了消息队列的方式进行业务解耦。

解耦后的POS退款、数据存储和性能都得到了极大提升。至于原始的收款数据怎么关



联退款呢？只需要原系统一个订单查询接口即可，在发起退款时先查询再同步。这样一来，订单收款数据库变成一套简单、庞大且易维护的集群，只需存储订单收款数据，同时提供历史数据查询即可。而退款只需要一套简单的数据库即可，且退款系统无需考虑订单的历史数据问题。

4. 资金归集系统——货到付款的掌门人

低效的电汇——请你“圆润”地离开。京东作为中国收入规模最大的电子商务公司和最大的零售商交易平台，每天会产生数量巨大的订单，这些订单货品通过京东物流配送到每一个用户手中。浩瀚如海的订单中，必然会有一部分用户选择货到付款，这一部分货款每天通过电汇的方式流转到集团财务部门。这种方式不仅低效、不安全，而且银行的手续费高昂，迫使技术人员对货到付款的资金归集方式进行升级改造。

利用互联网思维来思考，技术人员决定从以下三方面需求来着手改造：站长可以通过网银直接向财务部门转账；财务部门相关业务人员可以监控各站点转账情况；自动代扣，通过定时任务自动从站长卡中扣除所需转账金额。

资金归集——秀出你的肌肉。资金归集系统分为四个模块，分别是资金归集前端、核心交易系统、银行卡管理以及用于交互的银行前置系统。四个模块之间完全解耦，互不影响，相互之间通过HTTP调用。后续会升级接口，通过JSF进行接口的调用。

其中与银行交互的前置模块是最烦琐的，因为涉及与银行系统的交互，所以需要使用不同类型的安全证书，并且需要对敏感字段做加密处理。资金归集系统架构如图5-62所示。

◆ **资金归集前端。**主要用于与业务人员进行交互。如站长进行银行卡的签约、解约以及手工转账等操作。

◆ **核心交易系统。**主要起到纽带的作用，是资金归集前端模块、银行卡管理模块以及银行交互模块三者调用的纽带。

◆ **银行卡管理。**主要用于对银行卡相关的操作，如银行卡信息的持久化。

◆ **银行前置系统。**主要用于与银行系统进行交互，银行卡的签约、解约、转账等操作，最终都是通过这个前置模块与相关的银行系统进行交互。

站长通过资金归集前端系统发出请求，如签约、解约、转账等。根据站长请求，通过HTTP调用核心交易模块的相关服务，而核心交易模块会再根据业务逻辑，通过HTTP选择调用银行卡管理模块或者是银行交互模块，最后将相关处理结果返回给前端系统，展示给站长。

在核心交易模块还有一个Worker模块，用于定时任务的执行，主要执行的任务有：自动代扣定时任务、交易查询任务、交易对账任务。

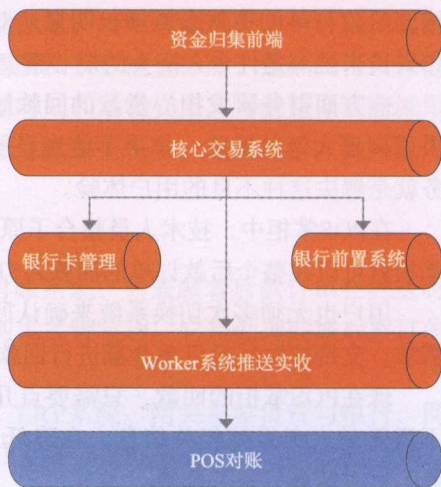


图5-62 资金归集系统架构

◆ **自动代扣定时任务。**自动从签约卡中扣除需要转账的货款。

◆ **交易查询任务。**通过调用银行接口对转账结果进行查询。这是因为银行转账结果可能是处于处理中，那么就需要询问转账的最终结果是什么。

◆ **交易对账任务。**对站长转账金额进行对账。

资金归集——请开始你的表演。资金归集系统的上线，颠覆了传统的线下货款回笼方式，极大地提高了货款回笼效率，保证了线下货款能够安全、稳定地归集到集团账目下。

自资金归集系统上线以来，站长再也不用每天急匆匆地跑向银行，财务人员也不用为站点是否成功转账而担忧。只需进入资金归集系统，鼠标轻轻一点，立刻一清二楚。

5. 价格保护——客户的定心丸

随着电商业务的快速发展，促销形式多变，导致商品价格波动频繁，有很多时候商品还在路上或者刚刚到手，同样的商品，价格已经发生变化。随之带来的是大量的用户电话客服申诉或表达不满。并且由于《消费者权益保护法》规定电商需进行7天无理由退换货，针对此项规定，客户可以选择拒收或者收到商品后选择退货，然后按照新的价格重新购买。对于用户而言，只是收货的时间延后了，而对于京东则需要承担物流的运输，库存占用等人力、资源的成本。

基于这些场景，价格保护系统应运而生。用户在京东上购物不再担心价格变化，并且可以通过自助方式申请商品差价赔偿，秉承着客户为先的理念，提高客户购物体验，满足用户需求。

同时还可以为京东节约物流成本，减少电话客服的话务量，避免库存的重复占用，提升京东的品牌认知度等。客户放心，技术人员省心，这是一个双赢的结果。

价格保护是一个从无到有，从最开始的只能靠客服人工计算，到后来的系统计算，再到后来的用户自助申请。价格保护从最开始的很少人知道，到现在变成了一种新的促销方式，中间经历了无数次的迭代，也经历了无数次的讨论。

从最开始的方便客服人员使用到满足每一位客户的需求，这一路经历了太多太多。最初上线价格保护系统只提供了京东主站和电话客服两个入口，但对于客户而言都需要花费一定的时间，尤其是在电话客服的时候基本都需要10min左右的时间才会申请成功。随着移动互联网的快速发展，大家使用移动端频率越来越高，价格保护系统也快速适应变化，快速入驻京东App端，用户可以进入京东App快速申请价格保护。

从最初的只能在收货之前可以申请，到后来的收货后一段时间内可以申请，再到后来的部分商品可以实现一个月甚至更长时间的申请。满足618期间不用再等着凌晨下单享受最低价格，而是在活动开始之前就可以下单，然后当活动开始的时候，如果发生降价就可以申请。价格保护系统把只有在618才能享受的优惠变成了天天都可以享受，不必再熬夜等着那一刻的到来。

未来，价格保护系统还有很长的路要走，也还有很多可以提升的空间，技术人员始终坚持着为客户先，始终为客户提供更好更加优质的服务为己任，不断追求优质的客户服务、愉快的购物体验。



6. 退款——说曹操，曹操就到

对于在京东买过商品并且退过款的用户来说应该不陌生，它的存在让客户在京东购物更加放心、更加安心。客户在京东商城下单后，由于种种原因不想要下单的商品，这时候如果没有收货可以选择取消订单或者拒收。另外，如果已经收货，可以通过售后申请退货，商品所支付的款项则由它退还给用户。它就是退款系统。

退款系统保障不少退客户一分钱的同时不让京东损失一分钱！退款系统的自动化保障客户越来越快地收到退款，正所谓“说曹操，曹操就到。”

◆ **资金的守门神。**资金安全一直都是京东重点关注的，每一笔订单都是无数客户和京东人共同创造的，每笔金额都无比珍贵。针对退款金额的准确性，技术人员做了一个单独的控制系统，保证每笔资金计算得准确。京东的支付方式灵活多样，给客户提供了无比丰富的支付手段，这对退款来说是个很大的考验，要将每个SKU的金额均摊到每笔支付方式上，总额还必须保证不缺失精度。针对订单不超退，技术人员也通过控制系统来做统一管理，对于资金从退款系统流转出口系统处，技术人员还有一道控制拦截，即使上游发生不预期的异常，技术人员也能卡住超额退款的单据。

◆ **麻雀虽小，五脏俱全。**退款系统的主要功能有两点：计算退款金额，决定退款途径。

退款系统会将用户支付的金额以及扣款项（比如运费、发票税点、全球购税费、安装费等）拆分到每个商品这个最小粒度上，然后根据商品品类，商品数量计算出实际退款金额。然后，会根据用户支付的渠道以及客户的意愿，决定退款途径，退款途径在保证退款合规性的前提下满足用户的需求。比如客户用京东余额支付的，但是客户却想把钱退到自己的银行卡中，这时候可以选择退银行卡。但是假如用户是用信用卡支付的，则只能退回到所支付的信用卡中，不能更改退款方式。

根据功能将退款系统分为三大块：支撑数据查询，页面展示部分；数据服务部分；任务调度部分。各部分再根据各功能的使用人数、使用频率，估算出最合适的服务器数量，做到资源利用最大化。不仅保证了各系统之间不会相互干扰，还保证了资源的非必要浪费。退款系统交互图如图5-63所示。

技术人员将系统数据分为热点数据和历史数据两种。将系统中不变的数据结转到历史库、清理数据库中冗余数据、保证数据库持续高性能输出。

退款系统处理单量呈指数级上涨，由原来的日均几千单发展为如今日均近百万单，京东订单的退款全部由退款系统自动进行退款。由原来纯人工处理发展为99.9%的系统自动化处理，大大降低了人力成本。

现在用户下单后支付渠道灵活多样，相应的退款途径也在逐渐增加，为了方便管理退款途径，退款网关应运而生。退款网关作为退款资金出口系统，是处理京东所有付款方式对应的退款方式统一出口平台。此系统承担了商城删单后返还客户虚拟资产的重任，保证了618期间客户第一时间拿到优惠券等虚拟资产的时效性，大大缩短了客户二次购物等待时间。

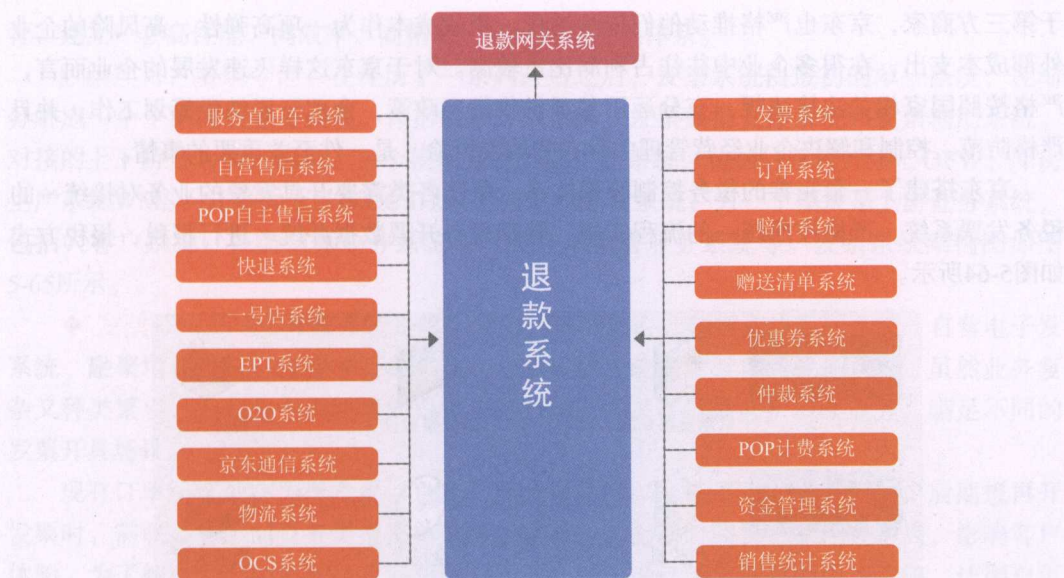


图5-63 退款系统交互图

◆ **流程可视化。**随着京东业务的逐步扩大，交易量有了突飞猛进的增长，相应的退款单量也在日益增多。目前日均退款单量为几十万单，618期间退款单量高达百万单。为了方便监控退款单量以及退款流转流程，衍生出了魔镜系统，该系统可以让财务管理人员更加清晰地监控每一个退款单的退款流程。在公司，财务人员可以通过公共显示屏上的饼图、折线图、流程节点图直观地掌握退款状态。在家，财务人员也可以通过移动办公软件监控退款单量、退款状态，退款的各种变化尽在财务人员掌控中。

◆ **海量退款数据——运用自如。**京东从诞生到现在，退款单量已达亿万级，为了保证这些海量数据的准确性以及完整性，退款系统采用大数据存储技术，将这些数据分布式地存储到各个集群中。保证在某个节点宕机等一些不可控的因素下不丢数据并可以及时恢复数据。还提供查询功能，在高频率地查询下，亿万级数据查询速度为毫秒级，高效快捷。

系统水平扩展的延展性，保证了618期间防止秒杀等场景后造成大批量删单的系统稳定。延展性也提高了公司资源的利用率。

◆ **独乐乐不如众乐乐。**对于客户来说，可第一时间拿到退款、优惠券等虚拟资产；对于京东而言，可提高用户二次下单率，达到双赢的局面。退款系统是京东逆向的最后一个环节，集资金安全、用户体验等于一身，资金的准确性、时效性是客户最为关心的，所以退款系统的稳定保证了客户在京东购物的安心和信心。

5.4.3 税控背后的故事

1. 京东税务体系概况

如何建立科学的税务控制和管理体系一直是京东乃至中国各大企业面临的一项严峻挑战。京东一向高度重视税务工作，在充分享受国家税改红利的同时，也积极履行纳税义务。京东以自营为主，对出售的每一件自营商品都为消费者开具发票，严格依法纳税。对



于第三方商家，京东也严格推动他们依法纳税。税收成本作为一项高弹性、高风险的企业外部成本支出，在很多企业中往往占利润比重较高。对于京东这样飞速发展的企业而言，严格按照国家相关法律法规，充分运用各项税收优惠政策，合理开展纳税筹划工作，并且严格防范、控制和解决企业经营管理中存在的纳税风险，是一件至关重要的事情。

京东搭建了一套完善的税务控制管理体系，集团各类需要开具发票的业务对接统一的税务发票系统，严格按照统一的流程实施，管理税务开票数据并统一进行报税，报税方式如图5-64所示。

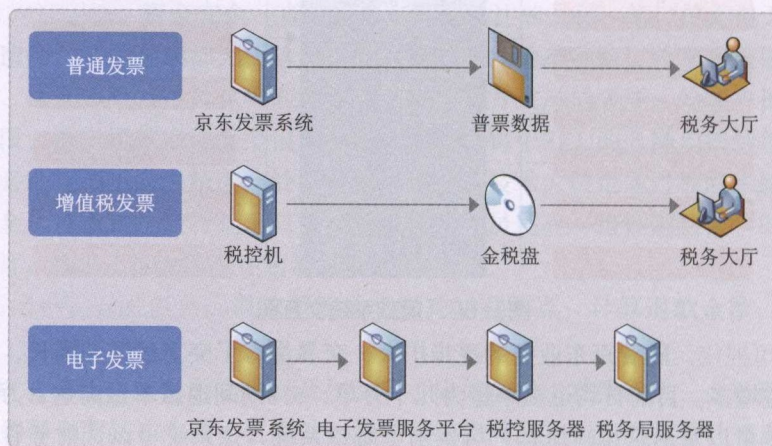


图5-64 京东发票报税方式图

承接的开票业务包括订单业务和非订单业务，个人、企业等用户业务，覆盖所有需要开具的发票种类：增值税普通发票、增值税专用发票、电子发票（电子版增值税普通发票）等。由上游提供原始数据，由税务发票系统控制统一的规则，记录每一张发票的开票数据信息，保持税务控制管理的高度集中一体化，实现企业利益最大化，规避风险。

发票系统作为税务控制管理体系的核心，在历经了一系列演变后，成功支撑了京东20多个业务的开票需求，包括商城个人用户、企业大客户、内部各结算系统、保理系统、京东云、京东E卡企销、备件库、维修中心等。

2. 发票系统的前世今生

历尽艰辛，坎坷前生。京东一直以来都坚持销售正品行货商品，自营商品均可以开具正规发票给用户，深受广大消费者的信赖和喜爱。作为订单生产过程中的重要环节之一，京东发票系统不仅是用户购买正品的凭据和报销必备材料，更是京东向税务局报税的重要依据，所以订单发票的开票效率、票面和金额的准确性与京东的客户体验和自身利益息息相关。最初的发票系统犹如一个新生儿，为京东而生，初出襁褓，摸爬滚打，摸索成长，只为满足京东客户对发票的需求。随着京东业务的爆发式增长，及越来越多的业务系统介入，发票系统不仅要满足订单快速生产的要求，还要尽可能降低开票成本，满足客户各式各样的开票需求。它不再是一个简单的发票系统工具，更是一个给亿万京东客户带来正规、安心用户体验的先行者。为了将技术做到极致，成为同行业典范，它需要一次脱胎换

骨，建立一套高性能、高效率、高精确、高可用的发票体系。

脱胎换骨，重现今世。在经历了一系列改造之后，发票系统由最初的单一系统，逐渐分解成一个个不同的业务模块，包括发票开票系统、发票数据支撑系统和发票辅助系统。对接的上下游分为两大模块：一个模块是上下游生产系统，包括订单履约中心OFW、库房生产系统WMS、订单拆分系统OCS以及逆向流程退款系统；另一个模块是上游业务系统，包括大客户系统、各类结算业务系统、各类外单销售业务系统等。发票系统架构图如图5-65所示。

◆ **发票开票系统。**开票系统是整個发票业务的核心，包括离线发票系统、自营电子发系统、融聚增票系统、普票非订单系统、ERP开票系统和发票自助服务系统。虽然业务复杂又种类繁多，但是整个模块规划得整齐有条理，能灵活地横向扩展新业务，满足不同的发票开具场景。

现有订单结算页在选择发票信息时，部分商品可以选择不开发票，而用户后期想再开发票时，需联系客服向财务提报工单进行发票补开，沟通时间长、操作成本高，影响客户体验。为了解决发票的补开以及发票咨询问题，团队设计了发票自助服务系统，让用户可以自主查看订单发票信息，了解发票开具规则，对未开票的订单自助提报开票申请，由发票自助服务系统将用户的开票申请推送至融聚开票系统，财务接收开票申请后直接进行发票开具，邮寄给客户。

◆ **发票辅助系统。**不同税局的发票，都有不同的发票模板和规则，包括发票的样式、限额、票面能容纳的商品明细条数限制、税率要求、备注信息要求等。发票模板管理系统的诞生是为了统一化管理各财务机构的发票模板，支持不同财务机构发票的精准快速拆分和模板生成。系统支持配置不同机构的发票模板信息并存放至内存中，并在开票前配置库房发票以及线下开票对应的发票模板关联关系，存放至Redis缓存中，保证订单数据进入到模板管理系统后，能够在个位数毫秒级完成发票模板的获取和订单数据的整合以及拆分。即使在618期间，系统性能依然能稳定在正常的水平。

同时，为满足不随货走的发票以及客户换开、补开发票的邮寄功能，发票系统打造了发票邮寄平台，支持各区域财务进行线上发票邮寄，包括自营青龙配送和第三方快递配送。开票系统完成发票开具后，会自动组装邮寄信息并推送至发票邮寄平台，省去中间环节中的人工操作，提高财务邮寄发票的工作效率。

全局主键生成系统是一个高可用、高性能的集群式的获取唯一主键的系统。发票系统数据量庞大，系统繁多，随着数据量和数据访问量的增长，数据信息的存储也相较以往发生了巨大变化，最明显的是分库分表技术的大量运用。将大量的数据储存在多台数据库机器以及多张表中，用于解决单表单库带来的数据库性能和存储空间的问题，利于数据扩展。但是分库分表带来的一个问题是，因为每条数据都是唯一的，必须被唯一标识，才可做业务上的其他处理。为解决大量高并发调用情况下获取唯一ID的性能问题、单点问题，实现高可用获取全局唯一ID服务并支持横向扩展，团队打造了全局主键生成系统。采用分表、A/B库、线程安全队列、左移运算、预加载理念，实现了系统的高可用性、高吞吐、高性能、可扩展性。

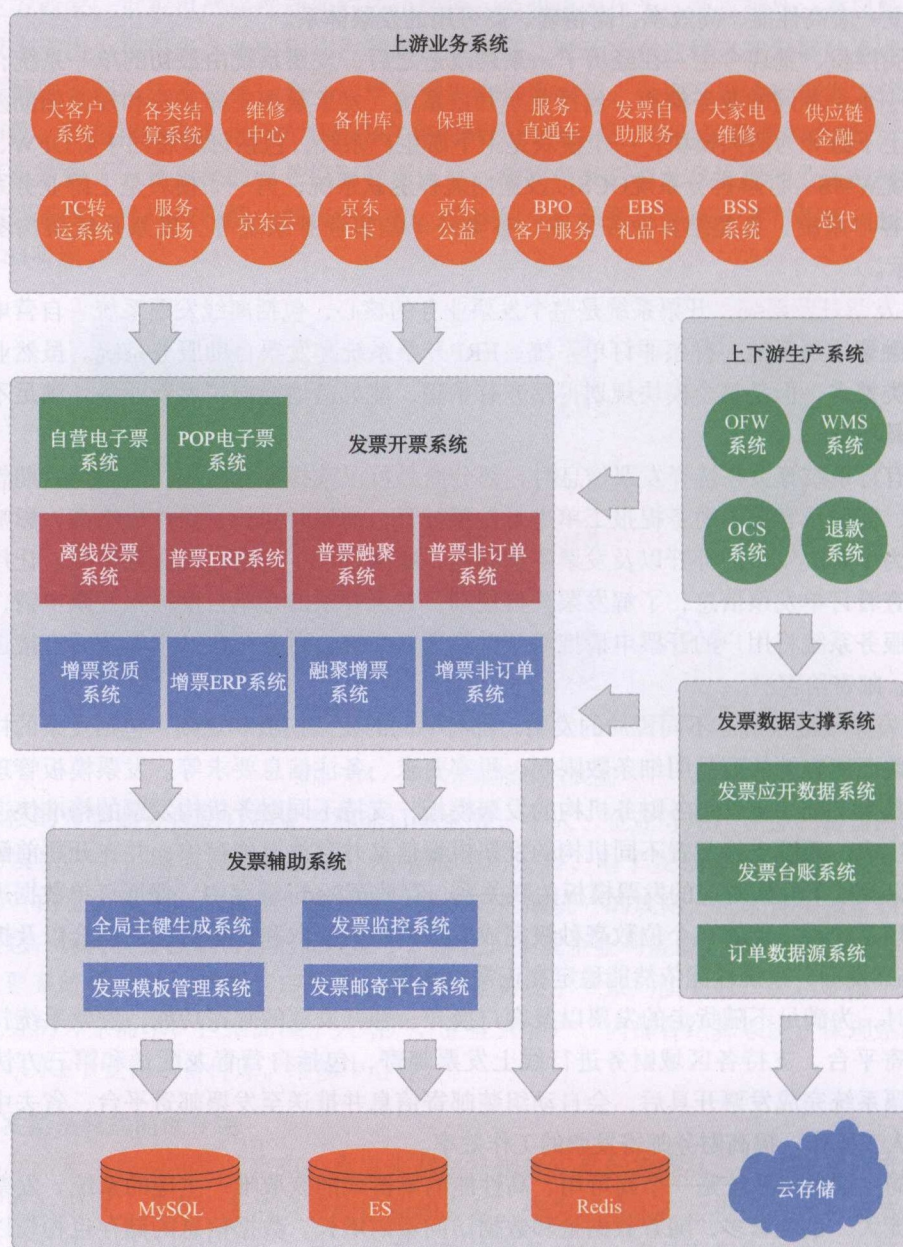


图5-65 发票系统架构图

◆ **发票数据支撑系统。**作为发票的核心数据群，发票应开数据系统、发票台账系统和订单数据源系统一起，为发票的各个业务奠定了夯实的基础。

数据支撑系统充分集成了京东的JMQ、JSF、JIMDB、云存储等服务，采用MySQL分库分表设计，支持主从备份以及主从切换，为发票系统添砖加瓦，保驾护航。

如今的发票系统经历了岁月的洗礼和战争的磨炼，早已成为了发票业务的老江湖。现今，我们迎来了从千单合开升级到万单合开的时代，满足大客户业务更多的开票需求，并

且从订单维度的发票信息存储升级到订单+SKU维度的发票信息存储，提供更细维度的发票信息查询服务，不仅实现了质的飞越，更实现了量的飞越。

5.4.4 智能资金管控

1. 企业资金账——一本黄金流水账

一本小小的家庭账本可以记录生活的点点滴滴，顺藤摸瓜，便于迅速掌握家庭财务状况；一本大大的公司账本可以涵盖企业全部资金流水，追本溯源，是企业资金账目管理的核心。京东也有这样一个账本，那就是资金台账系统。它记录了公司、供应商及客户之间的资金交易明细。每天面对几千万的交易流水，资金台账系统需要为各业务方提供稳定的服务，保证每条交易流水的实时记录，便于后续核对账本。资金台账系统架构如图5-66所示。

从业务上来说，银行资金账是财务所有核算的基础，是各种财务分析报表的核心来源。2016年，京东的GMV已从最初的1000万元增长至逾9000亿元，增速达90000倍，年复合增长率超150%。从交易规模来看，京东集团的交易额早已超过万亿元，巨额的成交量给资金台账系统带来了极大挑战。系统面临的不再是每分钟峰值几万笔的数据，而是每分钟峰值可达上千万笔。为保证系统稳定性，研发团队想到了拆分数据，但是拆分又会带来新的问题，新老数据要兼容，架构变化不能影响业务，分库分表的保存与跨时间区段的数据查询之间的矛盾，日常小流量与618峰值流量的权衡等问题。

保证系统稳定性是所有改造的重中之重，因此，技术人员决定将新老系统并行半年时间。技术人员有序地将老数据迁移到新的数据结构中，并进行定时校验，将老系统的功能慢慢切换到新系统。期间，老系统一直保持可用，这样技术人员就完成了一次类似“车辆高速行驶中更换轮胎”的操作。

财务业务要求的查询条件多，业务人员恨不得将数据库里所有字段都添加到查询条件里，这对技术人员分库分表来说很难处理。技术人员发现90%查询的是一周内的数据，8%查询的是一个月的数据，其余的查询，才会用到更早的数据。因此决定使用ES框架先去提升这90%的用户体验。通过ES框架，3个月之内热点数据的查询性能变得更好。数据异步导入ES，对ES的操作不影响当前系统，同时将时间作为索引，过期数据直接删索引，将历史数据迁入HBase，指定时间段查询。与此同时，技术人员也实现了跨库跨表系统查询，保证了在ES、HBase出现问题的情况下，业务依然不受影响。

618洪峰的挑战是严峻的，系统不仅要满足日常需求，更要经得起洪峰考验。资金台账系统在618期间表现优异，接口服务在618峰值单分钟几百万次的调用下，TP99维持在10ms以下，可用率保持在100%。

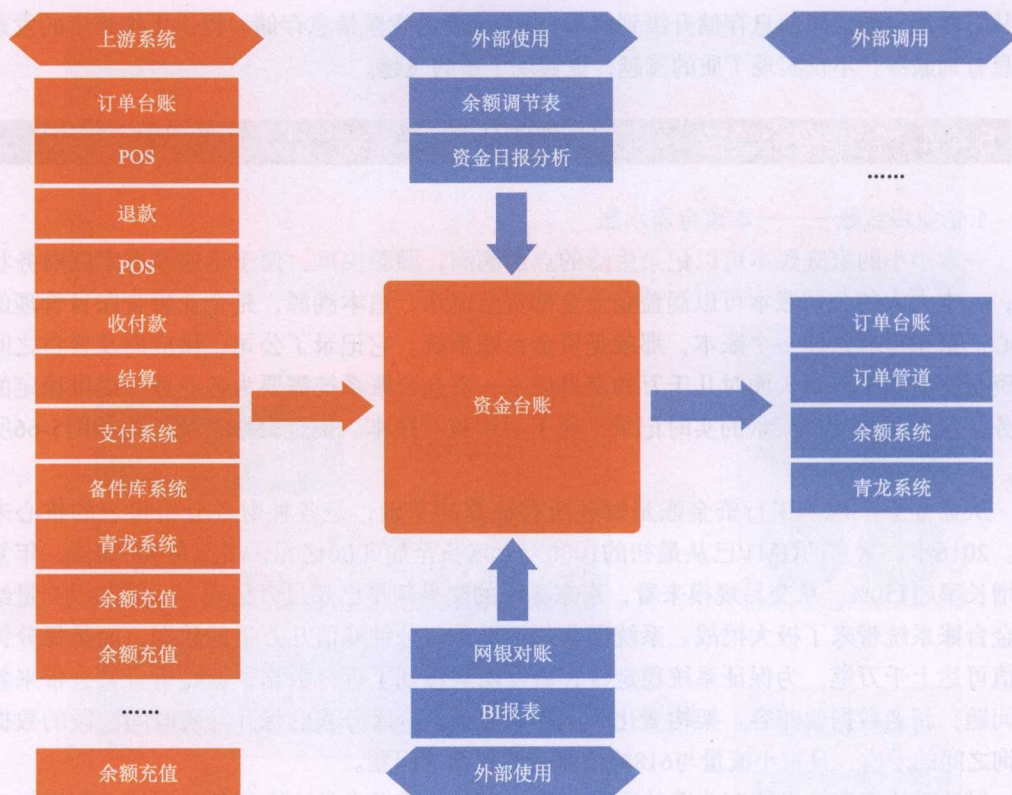


图5-66 资金台账系统架构

2. 余额调节表——海量资金流水对比

由于公司企业账（银行资金账）与银行交易产生的银行账（银行对账单）的入账时间不一致，因此会发生双方账面记录不统一的情况。为了防止出现差错，正确反映银行存款的实际余额，财务出纳每天需要将银行资金账与银行对账单进行手工核对，并编制银行存款余额调节表。由于对账工时长，出纳工作量远超过100%，并且线下Excel手工作业可承载数据量无法满足日常工作需要，故从节约人工成本和提高工作效率上来看，急需实现系统线上对账，自动生成余额调节表等功能。

余额调节表自动获取的对账单是以文件形式保存的，技术人员获取到文件后会对其进行解析，然后对账入库。618当天交易量激增，如果按照一般的解析方法，仅解析文件可能就会耗时几天，但是技术人员优化了解析规则，将单个文件拆分，让多台机器同时解析，从而缩短解析时间为6h。

3. 资金分配——不只是分钱那么简单

资金是企业进行生产、经营等一系列经济活动的基本保障。资金管理贯穿于企业整个生产经营的始末。所以资金的流转速度、准确性对企业发展起着至关重要的作用。

◆ **从收款说起——钱从哪里来。**京东收款业务分订单类收款和非订单类收款。订单类收款又包括商城订单收款（主要是在京东商城上下单的收款）和外单收款。非订单收款则

包括订单之外的所有业务收款，比如广告收款、配送员押金等。

京东收款主要采用集中收款和分散收款两种模式：集中收款，即客户通过收银台或其他转账等方式直接将款项转入江苏京东收入账户；分散收款，客户将款项转入京东各经营主体机构，包括京东分布在全国各地的分公司。

在京东集中收款模式下，京东收入款全部集中存入京东集中账户后，通过资金分配系统，将京东集中收入账户的款项根据实际经营收入划入各自经营主体账户，以明确划分各经营主体的实际营收情况，方便后续财务对各经营主体进行营收账款账务处理及业绩核算等。以京东商城上下单为例，客户在网站下单后，会生成订单号，同时通过客户收货地址匹配出商品出库的库房号和配送中心，由于客户付款的钱会统一先收入京东的集中收入账户，资金分配都会拿到订单商品出库的库房号和配送中心来作为依据条件，通过一定的规则匹配出这个订单号的收款金额并分配到对应的经营机构账户。

◆ **分块处理——与数据积压说再见。**资金分配系统会根据每个订单、每个生命周期的消息，按照一定的规则生成订单维度的分配明细和不同往来机构的结算单。

由于京东的业务不断增多，订单数量也随之增加。每个订单的生命周期里会有不同消息被资金分配系统所接收，如订单的收款消息、出库消息、退款消息、价保消息、完成消息等。由于接收消息不断增多，资金分配系统按照原先的两台机器单一处理接收到的消息，明显吃不消，会造成订单金额延迟分配的情况。所以技术人员进行了新一轮的改造，多台机器接收前端发送的JMQ消息，实时消费，防止积压；此外，对于接收的消息进行分块处理，不同的机器被分配不同的任务来接收这些消息。

◆ **Spark实时汇总结算单。**资金分配系统将前端消息通过一定的规则生成分配明细，这些明细需要针对不同的往来机构，对应的分配金额汇总生成结算单，也就是最后的调拨单，这样就能把资金打入各个经营机构账户。

随着业务的扩张，流水量的增加，按照之前的方式，直接使用SQL，在数据库中分类汇总得到最终结果已经不适合，一旦系统宕机，会直接影响京东财务核心业务。想要解决这个问题，技术人员有两种选择：一是对数据分块汇总再合并结果，这样做首先需要实现一个简单的分布式计算框架；再就是技术人员利用现有开源大数据框架。最终，技术人员选择了Spark，一个轻量级、高效的计算框架，将订单相关流水通过Spark Streaming实时计算，按照分类字段汇总成结算单，一天结束，结算单即可随即生成。



休远：财务研发部门总监，负责集团及各分公司的财税系统。通过一系列平台化建设，将结算、税务和资金等业务精细化、统一化、标准化和全面化。

第6章

赋能商家

生态共享，虚拟征程

赢得闲身共欢赏。为了让京东能够持续、高速地发展，第三方商家、虚拟业务等新的增长点应运而生。京东在稳固自营产品优势的基础上，正在不断努力让商家和京东共同成长。



6.1 开放平台营销系统

本节介绍的系统承担着大量基础性业务的建设工作，每个系统所代表的角色都是不可替代的，旨在解决如何保障开放平台业务健康发展、如何使平台上的商家高效便捷地开展业务、如何给消费者提供高品质的服务和商品等问题。

本节将对促销系统、订单系统、结算系统、商家系统的发展历程及技术亮点做详尽的描述。

6.1.1 促销系统

每逢618“狂欢节”，商家会发布大量促销信息，让利消费者。为给商家和消费者提供优质的服务，开放平台促销系统在从2011年到2017年7年的时间里，在“用户体验”、“系统架构”、“对外开放”三方面不断成长，磨炼出一身本领。

1. 用户体验

2010年9月，开放平台刚刚上线，当时只支持商家创建简单的单品直降。

2012年618前，为满足商家不同场景的营销活动，系统快速支持了创建赠品、搭配购、满减、满赠、满免等促销，店铺券也在那时出现。

2013年4月底，为解放商家，系统在创建促销时既支持选择商品，也支持选择商品下的SKU，并支持复制促销；为方便消费者自由组合商品下单，上线了套装促销活动页。618前压测发现同步促销性能不稳定，原因是多业务的任务数据都依赖同一张表，导致查询效率低，我们紧急将同步促销任务的功能迁移至任务驱动引擎，保证了商家促销即时生效；后来，将同步任务迁移至更加成熟的JMQ，使同步性能如飞一样快，商家创建促销时有了“丝滑”般的感觉。

2014年，为了降低商家在活动期间的沟通成本，活动提报系统在“女人节”闪亮登场。随着系统的不断壮大，活动提报系统已成为活动组织者最重要的工具。

2015年，618前我们不断进行创新：

- ◆ 微信手Q、App专享价。
- ◆ 在单品页、购物车、App店铺页展示可领取（使用）的店铺券。
- ◆ 店铺签到——提升用户黏性。
- ◆ 促销活动落地页——让消费者选购商品变得简单方便。

◆ 跨店铺促销——集合多商家优势商品联合营销，互赢互利。经过数次迭代，系统功能不断丰富；与活动提报系统的打通使得运营的工作效率大大提升。商家自组织跨店铺活动已经开始实施，将来肯定会有不止一个“啤酒+尿不湿”联合营销的成功案例。

◆ 京豆平台实现了发放、回收京豆，丰富了商家的营销工具；大促预告价使消费者能更放心地买到心仪的商品；区域促销可以满足诸如生鲜、家居等地域性商家的营销需求。

2017年初，“品牌商”系统实现了京东账号与品牌商会员、积分互通，在帮助线下品牌提升认知度的同时，会员权益也得到极大丰富。目前接入了朝阳大悦城、雅培、美赞

臣、山姆会员商店等，618期间会员实现100%增长。

2. 架构升级

营销系统主要经历了三次架构升级。

2014年3月开始搭建营销中心，将业务逻辑集中在一起，提供SOA中心化服务。618前商家端系统全部接入中心服务，系统运行平稳。

2015年618期间，促销数量一次次刷新纪录。面对严峻的形势，去Oracle迫在眉睫。不到4个月时间，完成了营销中心的改造：

- ◆ 利用Solr实现复杂的查询场景。
- ◆ 利用Redis（后升级为JIMDB）存储热数据。
- ◆ 利用JMQ异步完成业务逻辑。
- ◆ 增加降级自保护策略。
- ◆ 服务权限认证模块，防止未经授权的非法调用。
- ◆ 数据自检机制保证数据的一致。

在双11前完成数据的切换，经受了大考，为618大促打下坚实基础。在“给高速行驶的汽车换轮胎”的过程中没有出现一点问题，也是难能可贵的。

2017年618前，营销中心不仅承载了促销、店铺券，还孵化出京豆、白条、试用、砍价、预售等新业务。为了快速支持业务发展，营销中心拆分为促销、店铺券及孵化三大中心，并将原有的SSI基本架构升级为SI，进行独立部署。

3. 对外开放

2014年9月，营销中心基于SAF的服务调用框架，开放工作进展顺利。随着服务的增多，JOS的调用占比逐渐提高，高峰期达到30%左右。

2016年，京东与沃尔玛达成战略合作，山姆会员商店入驻京东，通过开放的JOS服务快速实现山姆商品双价格。

2017年618前，为提升商家和外部开发者对接营销服务的体验，降低接入成本，上线了新版营销API服务。

未来，营销中心会朝着“搭积木”方向发展，开放一个个“营销积木块”，届时按照规则即可搭建出丰富的营销工具，灵活多变，赋能商家。

每年618大促就如同一场战役，紧急需求总会层出不穷，而开放平台促销系统就像天降神兵，总会给商家及消费者带来更好的体验。

6.1.2 订单系统

1. 系统简介

订单系统是商家关注度很高的系统，为商家高效地处理订单提供支持。每年618，订单系统的压力都非常大。

订单系统的主要职责是：

为几十种订单业务线提供流程处理（大类型：SOP、FBP，小类型：全球购订单、



LOC订单、ECLP、供销等)；存储订单数据。为其他业务系统提供订单数据，如订单信息查询（包括查询订单基本信息、订单金额信息、订单发票信息等）、订单信息打印（打印面单、打印拣货单、打印出库单等）、订单生产（商家操作订单出库，录入物流单号等）、订单信息修改（修改运费、修改地址、修改快递单号等）。

订单系统每年都在飞速地增长，订单生成量和订单查询量每隔两年的变化，如图6-1所示。

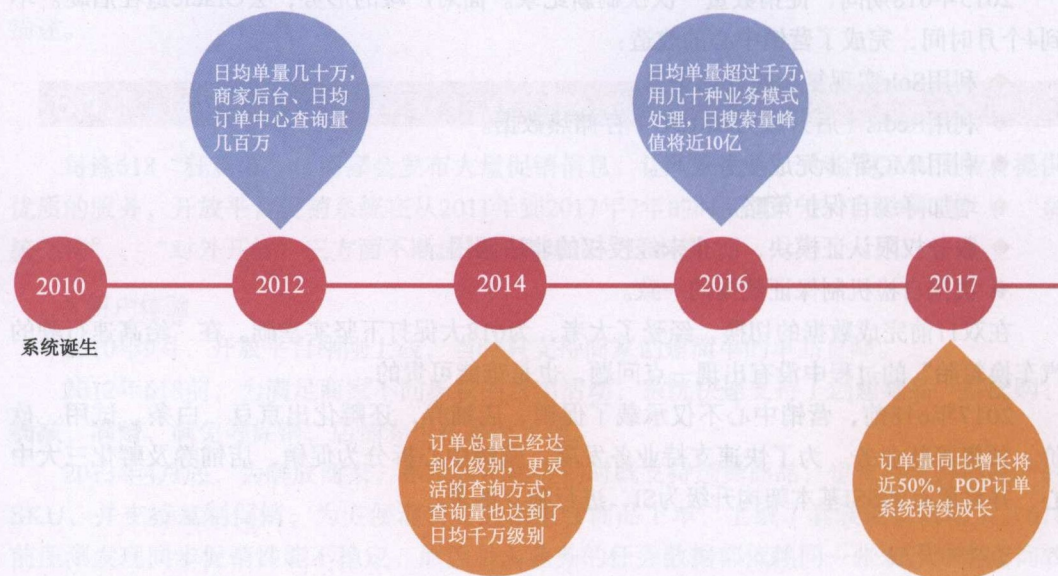


图6-1 订单系统历年简介

2. 订单系统的升级

POP平台系统从2010年到发展现在已经成为中国数一数二的电商平台系统，POP订单系统作为POP平台的一部分，致力于帮助商家更高效地生产和处理订单。无论是总订单量还是日均订单量，每年都在野蛮地增长，POP订单系统每年遇到的挑战也非常大，而最主要的挑战就是数据的存储和检索。我们为此也进行了很多次的优化和升级，订单系统的架构图如图6-2所示。

◆ **2012年：系统拆分。**订单系统于2012年从POP平台系统中独立出来，跟其他系统解耦，独立部署，分为订单中心、订单Shop系统、订单MAN端系统，所有系统通过京东的分布式RPC框架进行通信，还有一部分系统通过分布式消息系统对订单处理流程拆分解耦，通过消息机制异步串联。

◆ **2014年：订单中心升级。**2014年，POP订单业务突飞猛进，达到了亿级别的数量级，连接数据库的项目越来越多，查询的复杂度也越来越高。为了降低数据库的压力，更好地控制数据库的查询和写入，团队对数据库层进行了SOA化，所有读写数据库的操作直接在SOA层进行，有效降低了数据库连接，对数据库的操作控制粒度也更细了。订单的核心写入和读操作独立成OrderBase项目，删单申请等边缘订单业务独立成OrderAround项目，降低了非核心业务对核心业务的影响，对后续的去Oracle起到了重要作用。

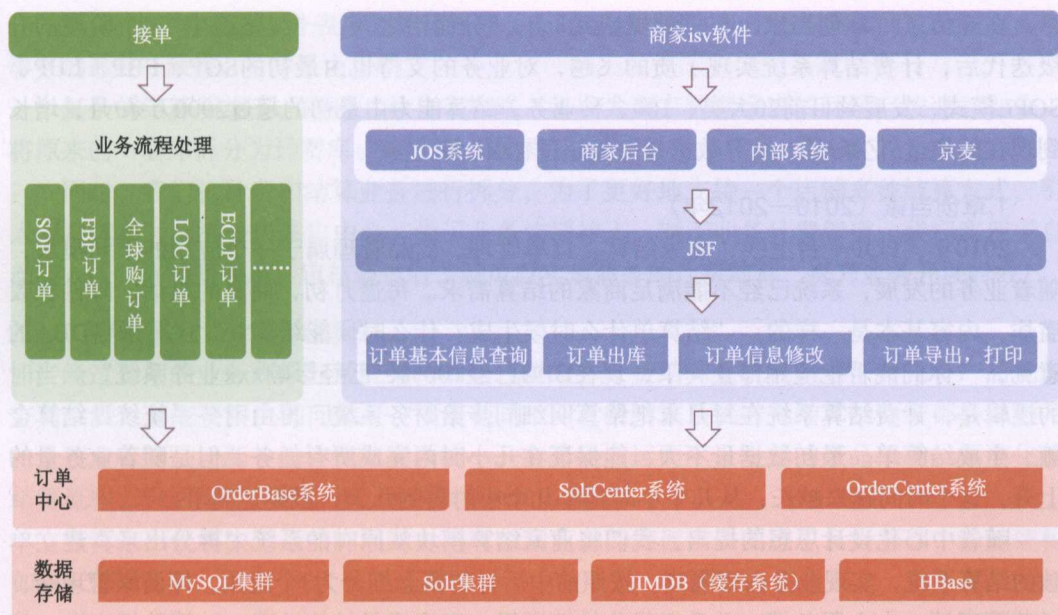


图6-2 系统架构图

◆ **2016年：去Oracle。**从Oracle迁移至MySQL集群，2016年，Oracle数据库的压力越来越大，很难撑过618大促，所以迁移至MySQL集群迫在眉睫。我们对未来几年的数据进行了评估，结合MySQL单表的承受能力，将数据分了1024个库。因为之前已经进行了数据库层SOA化升级，为迁移MySQL奠定了基础，因此可以不用修改上层系统，只需要在数据SOA层进行双写，同时进行全量导入，如果最终校验数据一致，就可以通过配置切换成功，将MySQL变为主库，性能提升显著。

◆ **2017年：Solr集群升级。**2017年的618突破千万级别，Solr集群作为为订单提供主要查询服务的搜索引擎，压力更是首当其冲。此次升级主要是对Solr集群按商家ID维度分了更多的分片，并且自主研发了动态路由模块、动态扩容模块，对Solr集群进行多机房部署，在Solr上层部署了机房流量路由模块，可以动态调整各个机房的流量。因为部署了多机房抗流量，流量可以被有效分流，我们开启了实时更新功能，将一些需要实时更新的功能也都切换到由Solr支持。通过这次升级，保证了618期间订单系统的稳定。

随着京东POP业务的迅猛发展，订单系统还会不断地完善和进步，未来系统会在更高效的大数据处理和智能方向进行更多的探索，为商家和用户提供更好的体验。

6.1.3 结算系统

电商行业风起云涌，业务形式变化繁多，作为服务于京东开放平台商家的计费结算系统，它的职责是给POP商家进行订单计费、结算，即计算与订单相关的费用明细，在结算账期内给商家生成结算单，完成结算付款和对账。因为京东商城交易频繁、数据量大，以及业务需求变化快，所以计费结算系统架构设计的核心是大数据量的存储与查询、繁多的业务快捷支撑。



在经历了“草创当家”、“神器威武”、“分而治之”、“涅槃新生”4个阶段的升级迭代后，计费结算系统实现了质的飞越，对业务的支持也由最初的SOP、FBP、LBP、SOPL模式，发展到目前20大类、100多种业务，结算能力由最初的超过2000万条/月，增长到现在的超过5亿条/月，结算数据量已经超过100亿条。

1. 草创当家（2010—2012年）

2010年，POP平台上线，商家结算、订单管理、商品管理属于一个应用的不同模块，随着业务的发展，系统已经不能满足商家的结算需求。每逢月初，商家的电话、邮件狂轰滥炸，内容基本是一样的：“结算单什么时候生成？什么时候能结算？”还有来自DBA的意见，“你们能否把应用停止一下，现在CPU已经100%，已经影响xxx业务系统”。当时的逻辑是，计费结算系统在每月末把结算明细同步给财务系统，再由财务系统统计结算金额、生成结算单。最初数据量不大，能保证在几小时内完成所有任务。但是随着业务量的上升，这个时间越来越长，从几个小时到十几个小时再到几天，最终达到了15天。

随着中心化设计思想的提出，我们将商家结算模块从原有的系统中拆分出来，建立单独的结算系统，实现业务内聚隔离。按照前中后台的概念细分为3个应用，即商家管理端、运营管理端、中心服务端，并升级优化结算逻辑，实现增量结算计费、结算数据查询、使用本地明细数据，不再依赖财务结算系统。同时升级数据存储方案，利用Oracle的二级分区方式，实现数据分区存储提升查询性能。

2. 神器威武（2012—2014年）

因为结算数据存储查询硬件由Oracle小型机升级到Oracle Extra Data一体机，所以在这个阶段结算系统满足了数据的快速增长。系统架构的重点也提炼出三维结算费用模型，基于费用模型对业务架构进行升级改造，让系统能够快速地支持新的业务。同时为了更好地利用Oracle Extra Data一体机性能，对4亿多的明细数据，进行“乾坤大挪移”，以商家ID和结算账期开始时间作业务主键，进行分区表结构改造，以结算账期时间作路由，便于数据的归档存储与查询。还利用MongoDB进行数据缓存，优化对账文件的生成。

费用模型有了，存储方案有了，剩下就是复杂的业务计费逻辑了。为此提炼出业务计费模型与计费业务领域特定配置语言（DSL），开展新业务时，只需要撰写相应计费逻辑单元实现，并配置计费模型，就可以实现新业务计费逻辑。同时实现了订单行计费逻辑，这样计费明细数据又无形地放大了好几倍，推进了下一步数据存储方案的优化升级。

3. 分而治之（2014—2015年）

当前系统数据总量100多亿条，月增长量5亿条，日增长量1500万条，大商家爆表。某充值商家一个月数据量达2000万条，某个月份计费数量曾达5.8亿条。面临几十亿的明细数据，再强的机器也有扛不住的时候，是时候分而治之了。我们使用以下几种方案：

- ◆ 设计一套存储归档方案，解决数据爆表、热表、事务处理问题。

- ◆ 构建数据存储归档工具集。封装实现数据库分库分表框架（db-sharding）支持读写分离、事务注解等功能，设计并实现数据同步归档迁移工具（data-sync），实现定时归档、延时删除、单条和全量校验等功能。

◆ 设计并实现分库分表事务处理框架（data-tx），通过事务任务重试回放机制，实现事务的最终一致性。

复杂的结算明细表，通过三种分表策略，解决大商家数据爆表的问题。按业务划分，将原来的一套库拆分为计费库、结算库、票据库。

同时，我们将计费和结算业务进行拆分，为了更好地支持一个店铺多种结算方式，引入账户体系，又拆分出账户中心。为了业务快捷接入，对于业务计费管道，我们实现了自动接入，只需填写表单，审核通过后，就可以实现业务的计费结算，提升业务开展时效！

4.涅槃新生（2016年以后）

经过长期的业务积累与系统架构的迭代规划，计费结算系统在朝着平台化、产品化的方向发展。首先要解决的问题是“所有使用POP计费结算系统进行结算的商家都必须再开一个POP店铺”，新一代计费结算系统——金算盘应运而生，全新的产品架构，标准化的结算流程，灵活对接各种业务，无须依赖POP店铺。

金算盘不仅解决了当前的问题，同时为新业务的快速开展奠定了良好基础，总体架构如图6-3所示。



图6-3 金算盘总体架构

业务的目的是提供服务并创造价值，系统通过技术模拟人的职责，更好地协作完成实现业务的价值。一个好的系统架构，就是一个好的组织结构，能更切合业务本身的职责，做好分工和边界。通过不同维度的划分和扩展，实现业务弹性增长。

6.1.4 商家系统

POP平台是一个开放式的合作平台，众多的商家在这个平台上把自己的商品销售给数



以亿计的消费者。商家系统作为POP平台的核心组成部分，是为商家提供基础服务的管理系统，涵盖了商家入驻、运营管理、数据分析等一系列商家服务功能，通过历年大促活动的历练，无论是系统的稳定性还是灵活性，都提升了许多。

目前，商家系统包括商家入驻管理系统、运营管理系统、主数据中心系统，通过三大系统构建了完整的商家运营管理体系。从图6-4中，可以看到商家系统的演进过程。



图6-4 商家系统架构的演变

1.孤星血泪（2010—2012年）

POP平台创立之初，业务场景以支持商家完成开店、发布商品等基础业务为主，所有的功能都集中在单一的系统里，商家系统的主要业务逻辑都集中在核心的领域层，以Spring作为中间层，通过iBatis去访问数据库，通过Struts来提供Web页面。

这种技术模式适合于早期业务快速发展但复杂度低的应用场景，模块间的调用简单（本地调用）、易于调试部署、水平扩展容易，由于业务量少，大促备战基本以值班观察系统为主，遇到问题时采用下线机器、重启Oracle等被动方案。

2.星罗棋布（2012—2014年）

随着POP平台的快速发展，各个模块在不同维度上的变化在加快，代码的耦合度也越来越高。随着时间的推移，情况越来越糟糕，多个模块的开发人员难以协作，不同模块容易发生资源冲突，任何一个模块出现问题都有可能拖垮整个系统，降低了系统的可靠性。这些问题的出现促使我们按业务划分成不同的系统，系统间的交互通过调用远程服务来实现，每个系统也可以对外部系统提供自己独有的服务。

我们把应用单独部署，使其运行在独立的进程中，可根据实际需求进行硬件资源的动态调配，使其有独享的资源（例如数据库），避免和其他应用竞争，职责单一，业务聚焦，每个系统由专门的团队维护，易于理解，容易扩展和变更，耦合度低，容易单独做技术升级和优化。

3.众星捧月（2014—2017年）

随着数据量和迭代任务越来越多，开发效率成为瓶颈。应用间通过远程服务或消息的交互难以调试；需要运维的系统更多，关联的系统越多，效率就越低，影响商家的使用体验。于是，我们进行了改进，把所有对其他系统提供的服务集中到统一的服务中心，系统

间的调用通过服务中心来协作，主数据中心提供了200多个接口，这些接口服务于600多个应用，平均每天的调用量就多达80亿次，遇到大促活动还要翻倍。在这样的背景下，系统在架构设计上的首要目标就是高可用、支持高并发并且不间断服务，在异常情况下合理地进行降级。因此，我们从以下几个方面做了升级优化：

高并发下的多级缓存技术。商家系统提供的店铺信息、类目、品牌信息会被下游核心系统（如购物车、结算页、财务系统等）依赖，这些下游系统的特点是并发高、要具备99.99%的可用性。对于这种场景，我们不仅使用分布式缓存，如Redis，还会使用如Google Guava这种本地缓存，以提高可用性和接口性能。

异常情况下的降级特技。系统跟机器一样，在高并发场景下会出现各种问题，例如访问突然剧增、网络超时等，此时，需要有各种降级策略来保护系统，主要分为自动降级、人工降级：

◆ **自动降级：**有些服务在一段时间内有可用率波动时可通过自动降级来减少故障时间，例如店铺信息读服务接口依赖的Redis可用率在90%以下的时候，会把读服务自动降级为本地缓存。

◆ **人工降级：**当出现严重错误的时候，例如在数据库宕机主从自动切换不了的时候会有个大开关人工一键降级，将所有的读库服务切到从库上，保障核心服务正常可用。

多机房部署与容灾。对于京东这种交易规模线上服务宕机都是以秒来计算的，宕机的每一秒都会成为事故，针对一些诸如天气、施工导致的线缆被破坏等不可控因素，也要有应对方案。商家系统应用依赖的资源，例如MySQL集群、Redis集群、ElasticSearch集群，同时部署在双机房，平时流量均匀分布，一旦出现机房网络异常，运维可以切换，同时商家系统通过降级系统可以从应用维度进行热切换，实现在1min之内恢复服务可用率。

隔离术

◆ **进程隔离。**应用按业务拆分后独立部署，每个业务单元都部署在不同的实例上，这样保证每个实例之间是物理隔离的，任何一个子系统出问题不会影响其他系统。

◆ **线程隔离。**我们会将应用内部的请求进行分类，不同的请求使用不同的线程。

◆ **集群隔离。**针对不同的业务领域，我们使用不同的集群进行物理隔离，例如店铺数据和画像数据分别在不同的缓存集群，任何一个集群出现问题均互相不影响。

◆ **读写隔离。**商家系统提供了不同维度的读写服务，读服务占比90%，写服务占比10%，例如通过Redis主从模式将读写分离。

升级后的架构如图6-5所示。

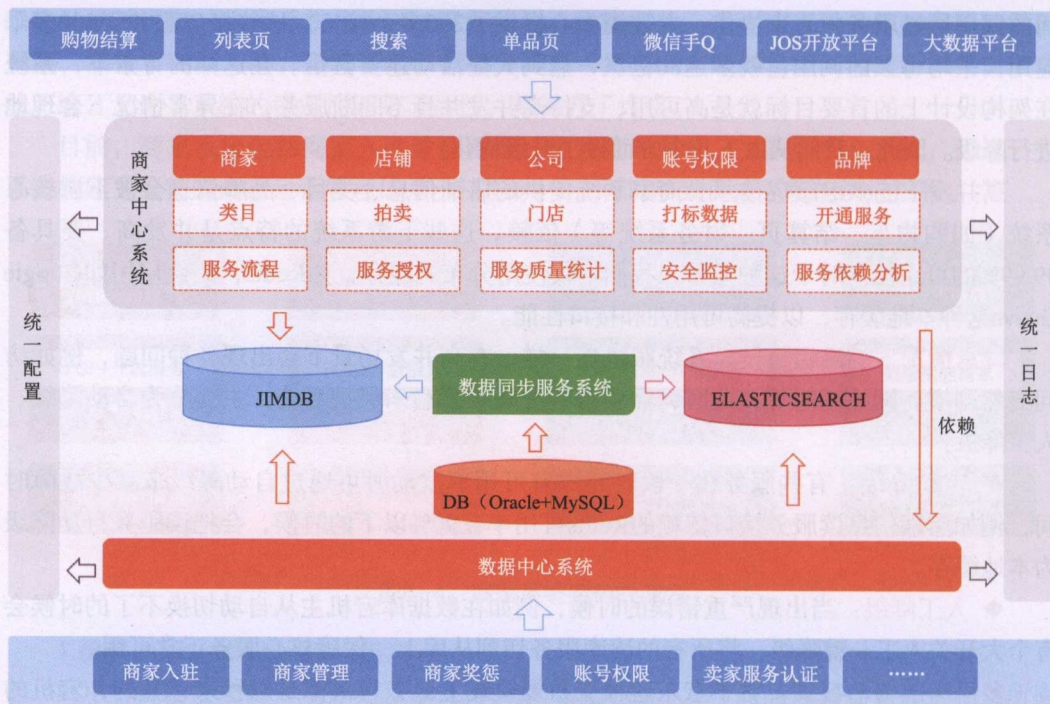


图6-5 商家系统的目前架构

4. 星光璀璨（2017年以后）

随着业务的演变，开发维护人员也从几人增加到几十人，随着代码量的增加，维护成本也在增加，多开发人员代码冲突问题逐渐增多，这种情况极大地破坏了代码可维护性，严重制约了开发人员的效率，并且加长了持续交付周期，单块架构并发开发各阶段工作高度耦合，同时单块架构生产成本随着功能和代码量显著增加而增加，图6-6显示了生产成本和代码量的对比关系，从中可以看到随着代码量的增多，成本越来越高。

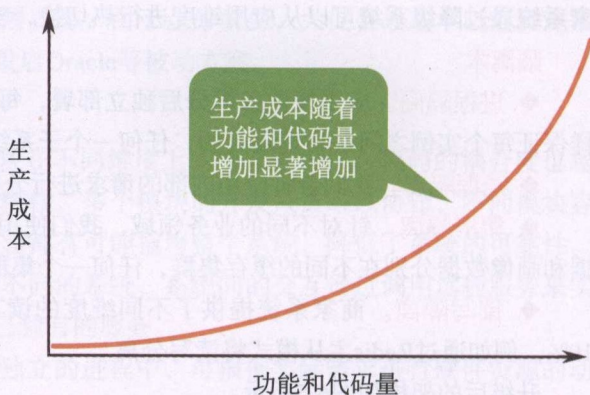


图6-6 商家系统生产成本和代码量比较

微服务架构正是解决这个问题的比较合适的方案。每个服务都是一个独立的业务单元，职责单一、代码库独立、服务进程隔离，当对某个服务进行改变时，对其他的服务不会产生影响；每个服务都有独立的测试机制，无须考虑破坏其他功能而建立大范围的回归测试；每个服务都是独立的进程，从部署角度考虑，服务与服务之间也是高度解耦的，这样能保障持续快速交付。图6-7展示了采用微服务架构进行并发开发时，各阶段工作是高解耦的。

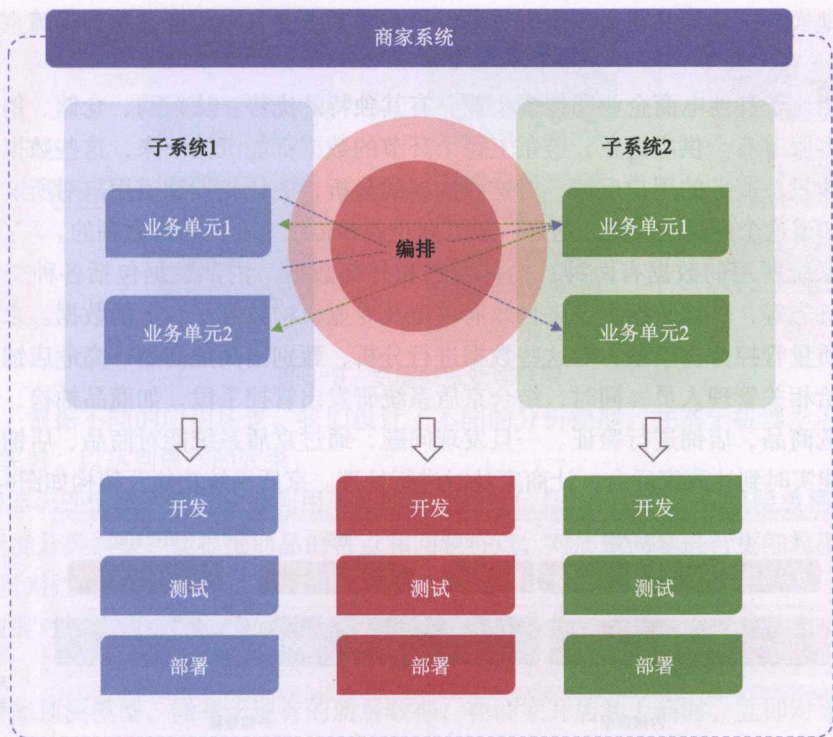


图6-7 微服务架构并发开发各阶段工作高度解耦

从单一架构到分布式架构再到微服务架构，商家系统架构不断进行迭代，随着业务的不断发展，商家系统架构必将进行新的升级。之所以以往的每一次架构升级都是耗时长、成本高的工作，是因为模块耦合比较高、代码冗余高。接下来要从业务单元化的角度考虑，让系统架构变得更轻便，灵活程度更高。



老炮：京东商城开放平台资深规划师，开放平台从无到有的见证人与实际参与者。
POPer：京东资深系统架构师，主导规划了POP开发平台的系统架构设计，参与了众多大型创新项目。

6.2 生态系统，保驾护航

6.2.1 京盾系统

京东视质量如生命，对假货更是零容忍。特别是当前既有自营、又有平台业务，商品质量更不容忽视。为了提高商品及服务质量，京东一方面在遵循和借鉴国际组织和企业推行的标准和方法，如ISO9000、全面质量管理、六西格玛管理方法等，另一方面在商品质量管控上有许多创新之举，如将大数据分析应用到商品质量管控这一场景上，提高管控工作的精确度和效率。京盾系统应运而生。

京盾系统是京东商城监控商品质量、管理商家风险的平台。通过红线抽检和风险调



查，有效地实现了对商品质量的全方位监控。京盾系统在京东内部被称为保障京东品质的安全卫士！

京东相比于其他电商企业和传统企业，有其独特的优势：从采购、仓储、销售、配送到售后、客服等有全供应链条，链条上每个环节的数据都能沉淀下来，这些数据中有大量关于商品质量及服务的用户反馈，对这些数据的分析，能让我们知道用户对所购商品的看法，从而知道这个商品的质量，这是一般电商企业和传统企业比较难做到的。

京盾系统所用的数据有两种：内部数据和外部数据。内部数据包括各种交易行为数据、系统日志等，外部数据主要来自政府单位及专业机构监管平台上的数据。京盾系统用专业化的质量管控分析模型，对这些数据进行分析，甄别出高危商品、高危店铺，并通过系统推送给相关管理人员。同时，结合京盾系统研发出管控手段，如商品抽检、风险调查等，对高危商品、店铺进行验证。一旦发现问题，通过京盾系统能对商品、店铺进行相关处理，并能实时到达商家后台，让商家及时进行处理。京盾系统功能及架构如图6-8所示。

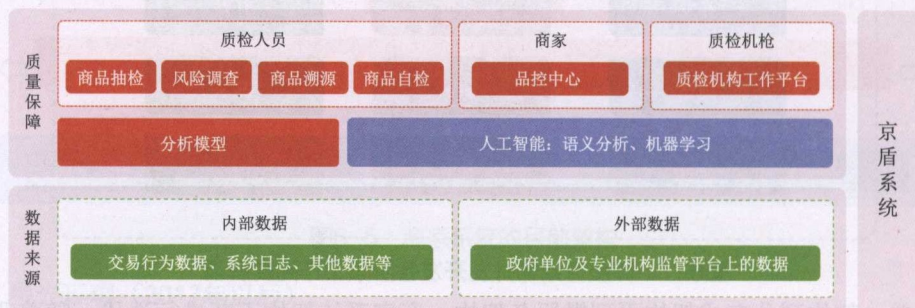


图6-8 京盾系统架构

京盾系统的成功，离不开大数据技术。在京东平台上，有数十亿的商品、数十万的商家和数亿的用户，根据平台规模产生的数据量与业务特点，量身订制了框架来支持系统的核心业务，如图6-9所示。在底层数据存储中，所有的原始数据都进入数据仓库统一存放在HDFS文件系统中，再按照应用需求分发到不同的数据集市进行数据加工与分析。

在数据分析与加工过程中，充分运用Hadoop平台提供的强大计算能力，在2h内就可以完成全部商品质量的计算与评估，并根据后续不同的应用场景，把质量指标分别存放于不同的系统。例如，把商品级别的质量数据存储于IMPALA系统中，借用IMPALA提供的并行计算能力来支持后续的实时分析；由于分项指标数量随着模型的变化而时常变化，根据HBase NoSQL的特性，把各种分项质量指标数据存放于HBase系统中，向质控人员展示更多的细节。

京盾系统之所以能提供高标准的质量指标，先进的质控理念、优秀的质量模型和超强的文本分析能力功不可没。首先，京盾系统处理的数据80%来源于非结构化的文本数据，能否准确地分析这些数据，直接决定了整个项目的成败。对文本数据的分析，采用了基于规则和使用深度学习相结合的框架，实现了最好的效果与最快的响应。基于商品类目训练出的深度学习（CNN，LSTM）模型可以提供95%以上的精确率和召回率，达到满意的效果。

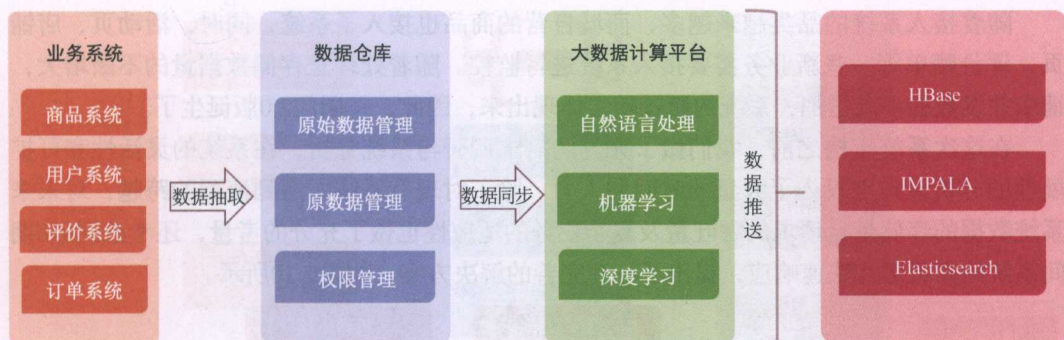


图6-9 京盾数据处理框架

同时，根据不同的应用场景，我们设计了不同的分析模型，并基于机器学习，不断地进行自我完善，以达到最佳的效果：

- ◆ 智能识别模型，能充分地利用商品的各種相关信息，识别出不同风险级别的商品。
- ◆ 智能分类模型，能根据商品的特点和问题特点，对质量问题进行更细粒度的分类，以进行有针对性的改进。
- ◆ 智能对标模型，能根据外界公开披露的一些质检信息，及时发现并锁定平台上相应的产品。
- ◆ 智能预测模型，能基于现有的质量数据，在商家开店和上新时，立即对该商品质量进行预测，提前进行质量管控，以达到预防的目的。

京盾系统2015年上线以来，经过两年多的发展，建立了一套成熟的商品质量分级标准，这套标准也可对商家进行分级，这为京东品控的对外开放奠定了基础。目前，通过跟一些业务部门进行合作，发现了很多适合这套标准的应用场景，如选品、信用评估等。

同时，在人工智能方面的探索会更多，为品控工作引入更多智能的元素，让工作变得更高效。

6.2.2 正阳门系统

正阳门系统是一个利用人工智能技术（图像识别技术、自然语言处理技术）对京东商城页面信息（图片+文字）进行合规性审核的系统。该系统主要通过人工智能技术自动识别商品信息、活动页信息、店铺页信息是否符合京东平台规范，检查是否含有虚假宣传信息、是否涉黄、主图是否合规、标题是否合规等。

正阳门系统每天对京东商城全站信息进行实时审核，如果发现违规信息会通过邮件、系统消息的方式及时通知对应的运营人员和商家，并提供违规明细，便于商家及时进行整改。对于经常违规或有严重违规行为的商家，会在后台记录该商家的违规次数，当违规次数积累到一定数值之后，会通过搜索降权、限制活动提报资格等方式对违规商家进行处罚。

正阳门系统最初是为了对商城POP商品的某几个品类下的商品介绍进行合规性监控而开发的。当时因需求紧急，正阳门系统只是做了MySQL数据库的分库处理，用Solr做条件搜索索引，便快马加鞭地上线了，系统上线后同时提供人工和机器双重审核。



随着接入系统的品类越来越多，商城自营的商品也接入了系统，同时，活动页、店铺页、评价晒单等一些新业务需要接入系统进行监控。随着处理且存储数据量的不断增大，请求并发数的不断增加，系统的瓶颈随之体现出来，因此，正阳门2.0版诞生了。

在这次系统重构之前，我们做了充分的需求调研与系统分析，在系统的灵活性和可扩展性的分析设计上投入了大量的时间和人力。除了对现有系统所遇到瓶颈的跨越，对未来系统数据的存储量、请求的吞吐量及复杂业务的适应性也做了充分的考量，还考虑到618期间系统的高可用和快速响应，提出了一套完善的解决方案，如图6-10所示。

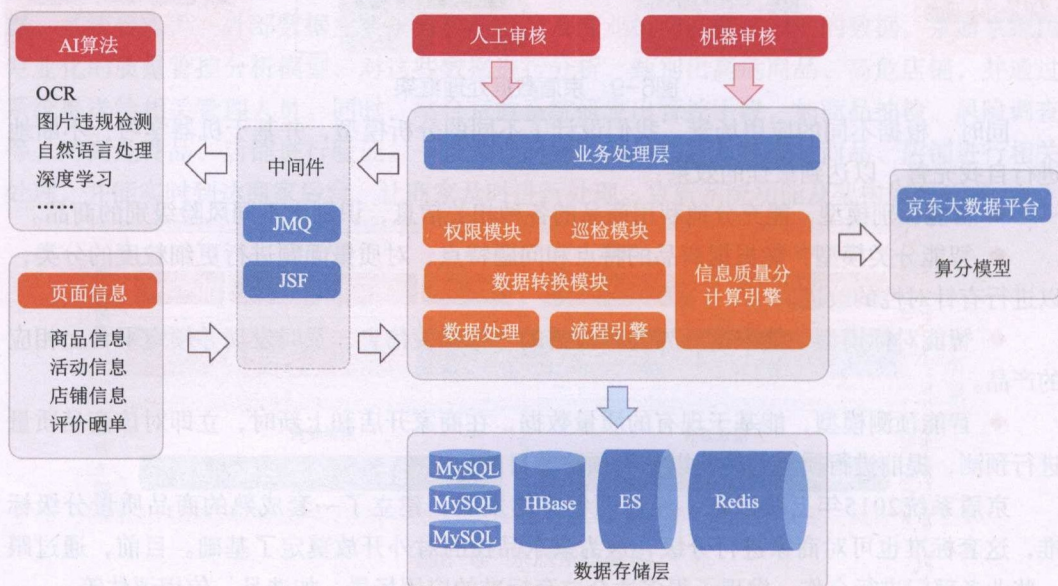


图6-10 正阳门系统架构

◆ **海量数据的存储与处理。**目前系统每天生产的数据行都在上亿的级别，并且这亿级的数据量都在不停进行读写操作，随着时间的积累，以及业务丰富度和复杂度的提升，数据量更是呈几何态势增长，所以在数据存储上我们对数据分别进行了水平和垂直拆分。数据库用到了MySQL和HBase，我们对MySQL进行了分库和分表操作，对HBase进行了Region预分区处理。同时选用ElasticSearch作为搜索引擎，以处理PB级结构化或非结构化的数据。选用Redis集群缓存热数据，以提升响应速度。

◆ **高并发请求处理。**目前系统每天需要处理的页面审核请求达2万次/s，为了降低系统压力，在处理请求时系统做了聚合和去重处理，这样可以释放更多的CPU资源，为后面的数据处理做储备。

◆ **AI的应用。**上面的讲述已经体现了系统需要审核的数据是海量的，如果将海量的数据交由人工去处理，那么所投入的成本是巨大的，所以在AI高速发展的今天，我们借助AI技术，实现系统审核线上自动化，由机器去判定信息的合规性，这不仅能大大降低人力成本，还能提升运行效率。此次系统应用了OCR文字识别、自然语言处理、图片质量及违规信息监测等多项技术，运用深度学习，不断迭代，不断优化模型，提升识别的准确度，为系统的高效运行提供了不可替代的保障，如图6-11所示。

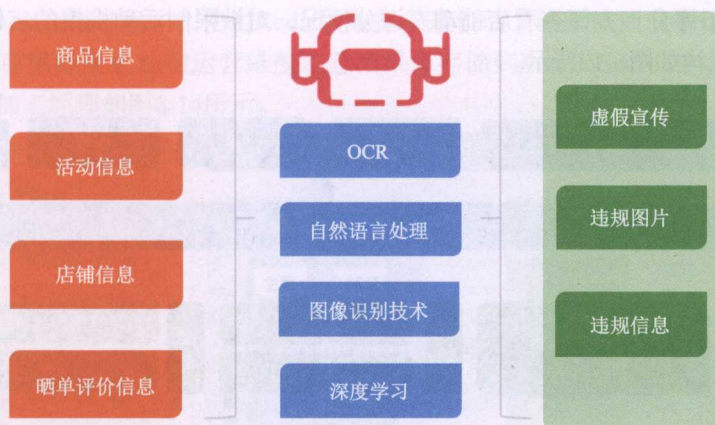


图6-11 正阳门系统人工智能技术的应用

正阳门系统上线以来，对商城所有页面信息的合规性进行检查，清除虚假宣传、涉黄、涉暴、涉恐等不良信息传播，大大降低了页面信息的违规率，为消费者营造了一个健康、可靠的购物环境，同时规范商品信息发布，为消费者营造良好的视觉体验。在618期间，更是加强对页面信息的检查力度，为大促的顺利进行提供了保障。

6.2.3 星盘系统

“客户为先”是京东企业文化核心之一，致力用优质的服务质量来创造极致的用户体验，而极致的用户体验需要商家通过售前、售中、售后各环节为用户提供优质的服务来实现。如何提高商家服务水平，如何提升服务生态整体环境，是整个电商行业急需解决的问题。

星盘系统是为部门负责人、运营人员、服务质量管理人员所提供的商家销售全流程服务质量的监控平台，能够全面掌握店铺服务水平，及时发现、及时推动解决异常问题，为商家提供具有针对性的帮扶策略及建议，完善商家管理机制；优化售前、售中、售后服务流程，降低用户投诉比例，提升商家服务质量，从而提升整个京东平台的商家服务生态环境。

1. 保障服务质量

星盘系统通过对咚咚、400通话详情、订单、服务单、纠纷单等底层数据加工形成咚咚满意度、400接通率、订单揽收率、售后满意度、交易纠纷率等售前、售中、售后服务指标，基于服务指标数据对店铺服务过程进行监控，根据京东平台规则对店铺进行奖励和处罚、对店铺服务质量进行监控和预警，同时通过服务中心接口为京东主站、天河之音、奖惩系统等其他系统提供数据支持。

运营人员通过星盘系统可以对所管理的商家店铺进行管控，全面掌握店铺的服务水平。主要功能模块分别为服务概览、店铺服务分、京东品质认证、店铺动态评分：

- ◆ 服务概览，展示店铺整体服务情况。
- ◆ 店铺服务分，明示店铺行业排名表现，准确快速定位服务缺陷。
- ◆ 京东品质认证，每个月的认证情况及认证标准清晰可见。



◆ 店铺动态评分，方便查看店铺动态评分情况，对被限制活动提报的店铺提前预警。
星盘系统架构如图6-12所示。



2. 技术驱动业务

离线数据加工。星盘汇集了咚咚、400通话详情、订单、服务单、纠纷单等维度数据，基础数据分散在各个生产系统，并且总体数据量在亿级别，所以，我们采用数据仓库抽取生产系统数据，并按照不同维度（主题）分区存储，在基础数据上建模，并拆分成不同层次的基础数据，下游数据集市同步数据仓库基础数据，做分类聚合计算，处理成星盘系统的日、周、月维度数据，推送到星盘数据库中，基于不同的数据量和不同的查询场景，我们将数据存储MySQL、HBase、ElasticSearch中，如图6-13所示。

◆ 订单维度数据结果集因数据量过大，所以存储在HBase中，并且参考kylin的rowkey设计原理，设计了一套rowkey索引表，存储在MySQL中，并且缓存到Redis中。

◆ 服务单和纠纷单数据量相对较小，我们通过MySQL分库分表存储。

◆ 对于部分场景百万数据量下载功能，我们将数据存储到ElasticSearch中，通过ElasticSearch游标分页查询，极大地提升了下载性能。

星盘系统几乎全部是读操作，为了提升读操作性能，系统做了大量优化方案，例如合并页面请求、数据库串行查询改为并行查询、数据查询结果预处理、页面延迟加载等，保证页面平均响应时间在1s内，大大提升用户体验。

实时数据加工。星盘系统的实时数据为运营实时决策提供了重要的依据。我们通过京东Kafka实时采集生产系统上报的消息，或者通过实时采集生产系统数据库Binlog获取实时消息，发送到Kafka消息队列中，下游通过Storm集群连接Kafka队列对消息进行实时聚合汇总，在聚合过程中关联一系列维表，维表数据缓存到Redis中，也缓存到本地，从而提升处

理性能。在Storm处理过程中，使用Redis做数据去重校验，同时按照商家ID和运营ID进行分发，最后将商家维度汇总和运营维度汇总结果集存储到ElasticSearch中，供前端查询使用。实时数据加工原理如图6-14所示。

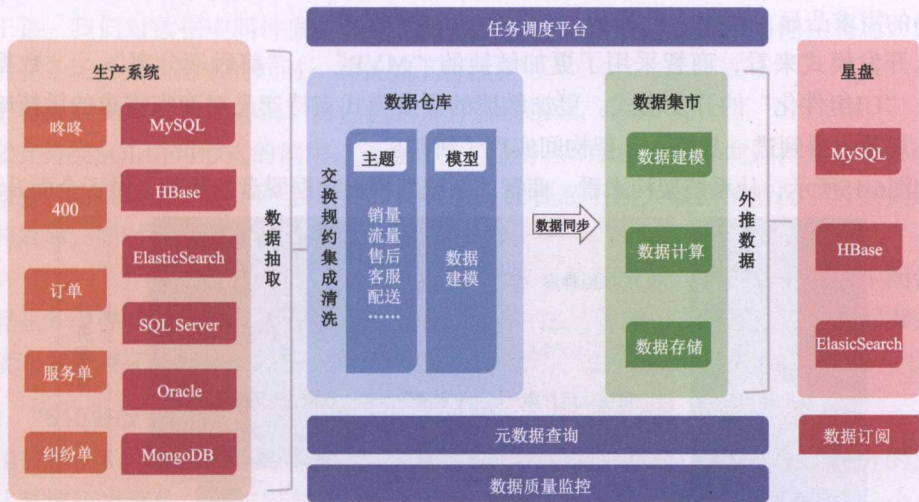


图6-13 离线数据加工

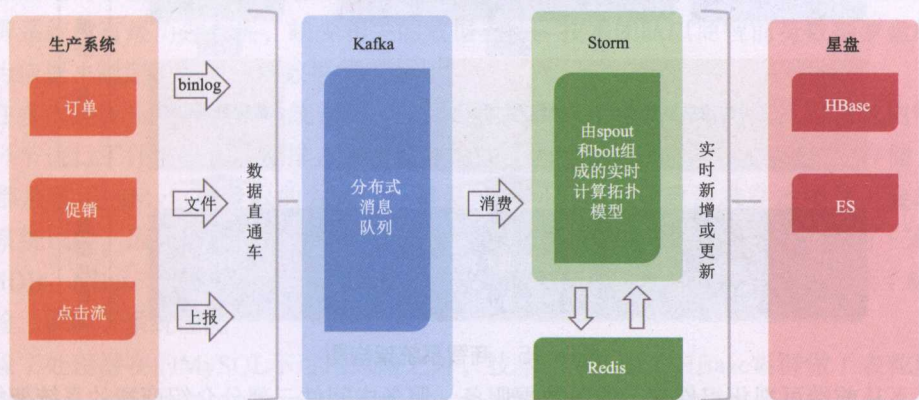


图6-14 实时数据加工

6.2.4 商智

京东商智（以下简称商智）是面向京东开放平台商家，提供运营数据、行业数据和营销支持的商业化数据产品。产品从基础的流量、商品、客户等角度切入，呈现店铺基础运营数据及其特征，提高电商数据化运营的效率。同时，商智向商家提供京东全行业的商品洞察、消费者洞察数据，赋能商家数据掘金。

1. 技术演进——从罗盘到商智

跟随京东POP业务的启动与发展，2012年初，POP数据罗盘启动研发与上线。数据罗盘1.0仅定位于满足商家基本的销售报表需求，产品设计得非常简单，系统开发模式与系统架



构也略显单薄。

2017年，随着商家数量的不断增加，数据量呈指数级上升，数据罗盘的技术方案也不断面临新的挑战。在数据化、智能化的浪潮之下，商家对于更清晰、更灵活、更智能的数据产品的需求凸显，商智应运而生。

从开发模式来看，商智采用了更加敏捷的“MVP”、“前后端分离”、“数据服务化”、“UI组件化”的开发模式。更加敏捷的开发模式支持产品对商家需求的敏捷响应，架构、规范也得到进一步统一，架构间实现了解耦。

如图6-15所示，从系统架构来看，商智的系统架构在数据罗盘的基础上做了全面升级。



图6-15 商智系统架构图

以下从前端可视化组件库、业务数据服务、服务中间件三部分介绍商智的系统架构：

前端可视化组件库。Grace组件库是基于Angular.js的前端UI组件库，包括一些数据处理相关的工具API方法，目前已积累了29个UI组件，如表格、日期、分页、查询等，同时支持多种皮肤，并且根据商智UI风格对整套组件定制了一套皮肤。Grace组件库提高了代码复用程度，保证页面类似交互具有很高的一致性，在很大程度上提高了开发效率。Grace目前已应用到多个数据产品中，如子午线、领航、供应商罗盘等，是一套比较成熟的前端组件化解决方案。

业务数据服务。业务数据服务系统将数据以元数据的形式拆分成数据源、指标、维度、属性、服务，同时接入数据中间件，可以满足大数据量、复杂业务场景的数据查询。同时使得接入方无须关心底层数据源，只需通过配置和标准化接口的形式即可获得多样化的数据查询服务。此外，业务数据服务还能针对多系统提供可重用的数据，保证数据的一致性。

服务中间件

◆ **数据中间件**。为了支持产品不断增长的用户量，以及越来越丰富的用户需求，在数据罗盘时代，数据中间件在面对大量的数据过滤、聚合等处理时，功能和性能都略显不足，于是，我们对数据中间件做了全新的改造升级。升级后的数据中间件在满足大数据量和复杂业务场景查询需求的同时，具备如下特性：

①支持TP99毫秒级的查询，可以达到百万级别QPS。

②支持类SQL语句形式的调用，规范统一了业务层对HBase的查询过程及结果获取。

③通过协处理器，以一种数据并行计算的过程提高了在大数据量情况下对数据过滤及计算的效率，同时保证了对HBase无侵入。

◆ **缓存中间件**。缓存中间件基于京东提供的统一缓存服务，封装后的缓存中间件更贴近商智业务缓存的使用，具备如下特性：支持缓存的Key管理；支持按业务清理缓存；支持同时接入多个缓存集群；支持主、备缓存的切换；支持注解和Util的快速接入。

2. “618备战”纪实：安内攘外、枕戈待旦

2017年618，是商智正式上线运营后面临的第一次考验。技术团队做了充分的准备工作，从系统压力评估，到扩容与优化、应急方案的制定与演练等，认真对待每一个备战细节。在制定商智整体的618备战方案前，技术团队需要预估618期间的流量及系统压力。由于商智系统没有经历过大促，缺少相关的数据支持，技术团队以商智前身数据罗盘的相关数据为依据，制定了各项预期数据指标。

在多次连夜压测，确认系统整体QPS满足压测目标之后，技术团队又对系统内部最薄弱的环节进行了仔细排查。在排查中，发现一些功能性数据目前只在MySQL中存储，且读取也都是采用的MySQL读取的方式，一旦系统访问量过大导致请求穿透缓存，会产生极大的系统风险。因此，团队针对此类功能性数据（例如权限角色菜单等）准备了三级缓存（JIMDB、HBase、MySQL），以及HBase和MySQL主、从存储切换，大大降低了MySQL的风险，保障了系统性能。

除了处理潜在的MySQL不可用风险之外，技术团队还针对HBase集群做了表粒度的切换预案，以防集群出现问题；对实时模块的请求添加了刷新时间配置的功能，以备实时系统压力过大时降低请求压力。为保障商家的使用体验，技术团队将所有的应急和降级预案，都放置到商智运营后台，支持一键操作、一分钟内生效，提高了系统的灵活性和健壮性，达到了“安内”的目标。

商智承载着商家集中获取各维度数据的需求，除了复杂的内部系统，商智对商品、促销等外部系统也有几十项的接口依赖，一旦外部接口发生性能波动，将不同程度地影响商家的使用，所以既要“安内”，也须“攘外”。例如，获取店铺实时关注人数的接口一旦失效，商家将无法实时评估618给店铺带来的关注效果。又如，商家权限接口一旦出现波动，商家将因为没有权限而无法进入商智，或者无法进入商智的部分页面，进而导致整个产品不可用。为保证商智在618期间为商家提供稳定可靠的数据服务，研发团队依照“梳理—压测—监控—应急预案”四部曲，对外部接口依赖进行了备战。

◆ **梳理**。从优先级、应急联系人、影响范围（菜单与业务级别）、接口文档、历史压



测性能报告、接口是否经历过大促六个方面，对所有外部接口进行梳理和分类，并整理成完善的文档，以备发生问题时及时跟踪定位。

◆ **压测**。对所有的外部接口进行统一压测。研发团队根据商智当前的QPS等系统压力数据，预测出618期间接口可能产生的压力值，然后同接口提供方逐一进行沟通，提出性能预期，并进行压力测试。对于不满足性能预期的接口，团队与接口提供方进行了沟通，在保障618的大前提下，接口提供方迅速地进行了对应改造和优化。

◆ **监控**。对所有外部接口添加UMP监控，并配置短信报警，当接口不可用或者请求量超出预估值时，系统将通过短信的方式通知相关人员，第一时间做出调整或者处理。

◆ **应急预案**。对四个高优先级的核心接口（商家信息接口、用户信息接口、商家经营类目接口、订购接口）制定应急预案，如图6-16所示。采用类似快照的方式，在每天接口压力最小时，预先抓取所有商家的接口返回值，并且持久化到本地HBase中。与此同时，在HBase上添加了一层Redis缓存，以提升访问性能。一旦接口不可用，可以在1min内将接口依赖切换回本地数据，最大限度地避免商家的使用受到影响。

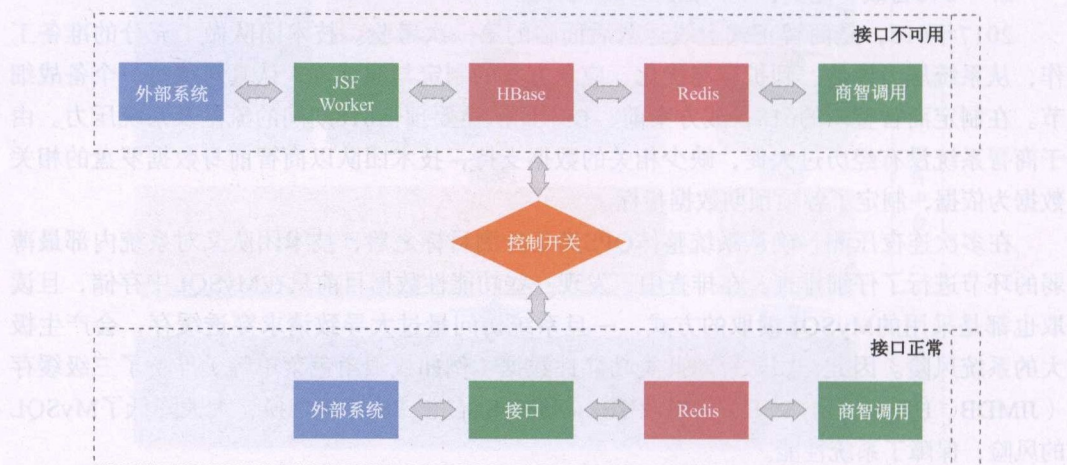


图6-16 高优先级接口应急方案及正常调用流程图

对于优先级较低的接口，制定了接口压力过大时通过人工控制接口的访问频次进行降级的应急预案。例如，实时流量与实时销售的相关接口，在原定的秒级间隔刷新的基础上，新增了可配置的开关。当接口压力过大时，可以人工将缓存时间延长，进而降低对外调用的频次；当接口压力下降之后，又可以人工将缓存时间缩短或直接关闭缓存，在商家无感知的情况下，动态控制调用压力，保障系统的稳定可用。

万事俱备，在618的前一天，北辰职场的技术团队派出精兵强将，驻扎亦庄总部集中备战。备战强度很高，团队成员脑力消耗也很大，在集团后勤提供免费自助餐的餐厅，商智技术团队大快朵颐，每个人都吃掉了双人份的美食。食堂大厨看到技术人员将食物一扫而光，自得地调侃说：“看你们这胃口，肯定是从北辰来的吧！”（北辰没有京东自己的食堂）大家笑着说：“咱们机器扩容备战，人也要跟着扩容嘛，必须要多吃一点！”带着扩容的机器与“扩容”的胃，技术团队枕戈待旦，实时监控各项数据，及时响应所有系统问题。最终，商智平稳度过第一次618的考验，为之后的不断提升打下了坚实的基础。

6.2.5 京麦

1.京麦介绍

京麦工作台（以下简称京麦）是京东商家的一站式生意管理平台，“京麦工作台”中的“京麦”意为“京东商家大卖”。2013年，当时的京麦叫作“京东商家助手”，是为了提升商家处理订单的速度而开发的，后来加入了商品管理功能。在所有人的印象里，“京麦”是“高效”的代名词。2014年，为了提升商家店铺管理的便捷性，京麦成功进行平台转型，更名为京麦工作台，接入了促销管理、数据罗盘、论坛等商家常用工具入口。2014年末，京麦发布移动App，搭建消息系统，为商家进行消息实时推送。在此期间，京麦不断对技术进行优化，通过多种互启协议实现“消息—操作”的闭环，迈出商家移动化办公的第一步。

自2015年起，京麦在不断丰富自身功能和系统性能的同时，在“平台”的基础上构建开放生态，努力为京东商家搭建一个良好的生态环境，如图6-17所示。“京麦网关平台”面向第三方应用开发者，提供API接口和相关开发环境。软件开发者可通过API获取京东用户信息、商品信息、店铺信息、交易明细信息并建立相应的店铺管理及营销工具。“京麦开发者中心”为ISV提供测试环境、技术咨询、产品上架、版本管理、收费策略等服务。京东商家可以在“京麦插件市场”购买增值服。在“京麦客户端”中，用户可以享受极致操作的体验，基于客户端的消息、资讯、协议、本地存储、稳定等特性，真正实现了高效快捷的一站式店铺管理。现在京麦正对交易能力、消息中心、资讯系统等系统进行能力输出及开放，让更多人通过京麦现有的能力快速服务于商家。



图6-17 京麦开放平台架构图



2. 架构演进

◆ **2014年：京麦开放平台诞生。**在2014年创建了网关系统的雏形，搭建了网关1.0系统，构建消息推送服务。使用Netty框架搭建了第一个版本的消息推送系统，形成了最早的京麦消息中心。这些为后面京麦的开放播下了一颗欣欣向荣的种子。

◆ **2015年：构建网关2.0，接口配置自动化。**网关API是京麦开放平台架构中的重要一环。每到618，API调用量会成倍增长，网关的治理和监控被提上日程。京东内部的数据分布在多个独立的业务系统中，系统之间通过JSF框架进行RPC通信。为了保障ISV在618期间稳定地使用这些数据，需要用统一的方式来整合这些接口。网关的API基于OAuth2协议提供，ISV通过HTTP协议接入开放平台的网关服务，调用期间要求确保将数据快速、安全地提供给ISV进行服务调用。在这样的高要求下，团队构建了网关2.0系统。

在2016年的618，团队对京麦消息中心整体架构做了一次升级改造，摒弃了原先的“半推半拉”方式，改为“半推半查”，如图6-18所示。推送的是消息命令关键字，推送量小，实时性高，做到第一步：快。查询的是消息实体并且消息实体从Redis存储改为ElasticSerach存储，消息不再像以前那样，拉取之后默认删除，而是当用户点击查看消息的时候才会在客户端显示消息的具体内容，做到第二步：准。同时，采集消息时按照消息进行分类处理，分别放入不同的JMQ队列，做到消息生产和消费互不影响。另外，团队改造了APNs服务，采用HTTP2的方式，解决了Channel异常及InActive时重连等问题，做到第三步：稳。

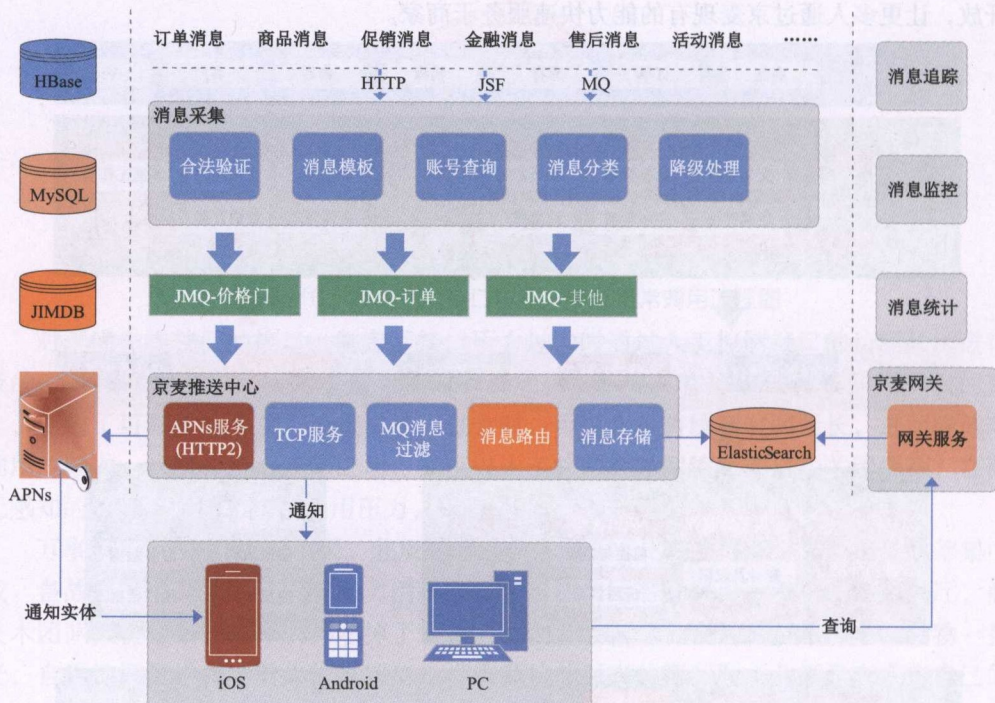


图6-18 升级后消息中心架构

在2016年618前夕，京麦插件市场上线，ISV的软件服务可以直接发布到京麦插件市

场，商家可以在插件市场中更方便地选择所需要的软件。京麦插件市场为商家提供了丰富的功能延伸，为商家的业务的腾飞插上了强有力的翅膀。

京麦从无到有，从弱到强，为平台上数百万商家提供了个性化、多样化的工具服务。帮助上百万商家提升经营效率，完成渠道的转型——京麦=卖家中心+京东咚咚卖家版+插件中心+更多功能。京麦定位为一站式、移动办公、开放式，其核心是为卖家整合店铺管理工具、经营咨询信息、商业伙伴关系。京麦的诞生，不仅提升了京东商城商家处理订单的速度，促进彼此间的合作共赢，更对电商行业进行了重大改进整合。同时，京麦技术架构正在不断扩大、优化，系统架构也会逐步承接更多的流量和面临更大的挑战。由开放平台、京麦构建的开放生态无疑将成为这个伟大事业的标兵，继续引领生态伙伴们向着新的征程进发。



iBD：商业数据产品专家，负责供应商罗盘、京东商智产品规划与研发。
xinbook：目前就职于京东商城，负责京麦工作台的架构与开发工作。

6.3 虚拟平台的618征程

6.3.1 智能调度出口流量平台

目前虚拟平台系统均为多机房部署，涵盖移动、联通、电信等运营商，每个业务系统还需要与多个商家进行交互。例如，话费充值系统对接23个商家、流量充值系统对接50多个商家，这些商家又分别覆盖移动、联通、电信等运营商。在这种情况下就会导致虚拟业务系统在商家交互时出现跨运营商的网络传输，而这种跨运营商的网络接口调用的成功率很难达到100%。

当商家接口调用可用率降低时，通常会通过联系商家或者由业务切换渠道等方式来解决，但这种方式不仅耗时耗力，而且不能解决根本问题，甚至问题会反复出现。尤其在618期间，单靠这种被动的解决方案势必会影响用户在充值过程中的体验。

为了提高业务系统与商家接口调用的成功率，我们自主研发了智能调度出口流量平台（Public Network Exit Route，简称PEER）。PEER系统会自动选择与商家相同的运营商网络的机房作为流量出口，以便提高业务系统与商家接口调用的成功率。

1. 推本溯源——抓住问题核心

◆ **跨运营商网络调用。**例如：京东移动机房部署的应用调用商家的联通机房，业务系统下游对接的商家较多，其中包含不同的网络运营商。因此，像这种跨运营商的网络交互，问题就比较多，接口调用的失败次数也会明显增加。

◆ **路途遥远。**例如：北京机房部署的应用调用深圳机房，像这种跨区域调用，中途会中转多次，从而造成网络调用失败。

◆ **不可抗因素。**例如：自然灾害导致整个机房不可用或人为因素误伤光纤等会造成接



口调用可用率下降，甚至整个机房都不可用。

2.推陈出新——智能匹配合理输出

智能负载均衡——客户端进程内负载。当客户端所在机房的网络运营商与商家所在机房的运营商属于同一网络运营商时，可以直接在本地调用商家接口，即客户端根据不同的商家选择不同的机房作为流量出口。通过智能匹配的方案，即可从根本上解决这些跨运营商或跨地域的问题，从而提升调用的成功率。

智能负载均衡——服务端负载。如果部署应用所在机房与商家应用所在机房不是用一个运营商网络，此时业务系统会通过PEER系统提供的接口，并通过JSF将流量导向与商家机房相同运营商的机房，这样就可以实现与商家在同一个运营商网络上进行接口交互，大大提高了接口的可用率。

智能调度流量出口机房。关于机房的智能筛选，我们会提供三个策略：

◆ 策略一：手动设置默认机房，手动调整，即为每个代理商设置一个默认的机房ID，保存商家信息。

◆ 策略二：根据实时可用率，选择一个服务性能最高的机房，通过监控平台根据key获取实时可用率，筛选出最近5min可用率最高的机房。

◆ 策略三：综合策略一和策略二，通过阈值&开关控制，避免网络抖动可用率不稳定，可能会造成商家的不良切换。增加一个开关和阈值，当开关打开时，默认机房可用率不低于阈值即可使用，低于阈值再通过监控平台筛选出一个可用率最高的机房。

通过这三个策略，可以真正实现整体机房的智能调用，为系统服务的稳定性提供了坚实的保障。

智能降级。引入熔断机制，优雅降级，避免某个业务线或某个接口出现问题后长时间占用系统资源。

智能平衡流量。通过权重平衡每个机房的出口流量。

3.初见成效——安全稳定系统保障

PEER系统引入Hystrix开源框架，将各业务线及各接口隔离，这样可以避免因某个业务线或某个接口可用率下降而影响其他的业务或接口。从业务指标上来看，系统服务的可用率已由曾经的99.9%提升到99.999%，机房的切换过程也由原来的2min提升到1s。

PEER系统采用多机房部署，可实现当某一机房网络出现问题时快速将流量切换到另一机房。同时，PEER系统支持部署多个集群，可为某一订单量较大的业务线单独部署一个集群来支撑，为业务系统不间断地提供服务。

目前PEER系统不仅为话费充值、流量充值、机票等各大系统保驾护航，也为各大系统的大促及订单飞涨提供坚实的保障。

4.遍地开花——让系统和用户快乐起来

经过了618及各种大促的重重考验，PEER系统不仅做到了提高接口调用的成功率，更在很大程度上让用户的购物体验得到提升。无论如何，用户满意才是我们最想要的结果。

以话费充值系统为例，与商家的接口调用TP99由原来的超过2000ms降低到500ms以

内，同时与商家的接口调用成功率也由原来的90%以上提高到了100%，这种情况下即为用户真正实现了充值光速到账且百分之百成功的最优体验。从单交易负载测试结果上也可以看到，在单机单实例下，并发用户数为30人时，TPS为3946笔/s，平均响应时间7ms。

有了PEER系统，维护人员再也不会因为接口可用率而夜里起床排查问题了，也不需要过多的人员为可用率的问题而伤神劳心。PEER系统不仅省去了烦琐的事务，也让系统更经得起考验，变得更健壮。

6.3.2 虚拟业务黄金交易流程

不知常常购物的小伙伴们是否发现一个现象：想要预订一张飞机票、购买一个旅行箱，却不能在一个订单中结算，而是要分开下单、结算。这是为什么呢？

答案就是虚拟商品与实物商品存在着极大的差异，京东倾力打造多年的实物交易流程无法完全满足这些不需物流配送、时效性强、更注重服务体验的虚拟商品的需要。所以，虚拟商品需要特殊处理。研发部组建攻坚小组，以虚拟商品中心为基础，以订单和促销为核心支撑，以支付组件、退款网关、结算、数据集市、用户中心等公共组件和服务为重要辅助，搭建起一套便于复用、易于扩展且符合虚拟业务特性的黄金交易流程。

虚拟业务黄金交易流程技术支持和整体架构，如图6-19所示（虚拟基础平台整体架构）。

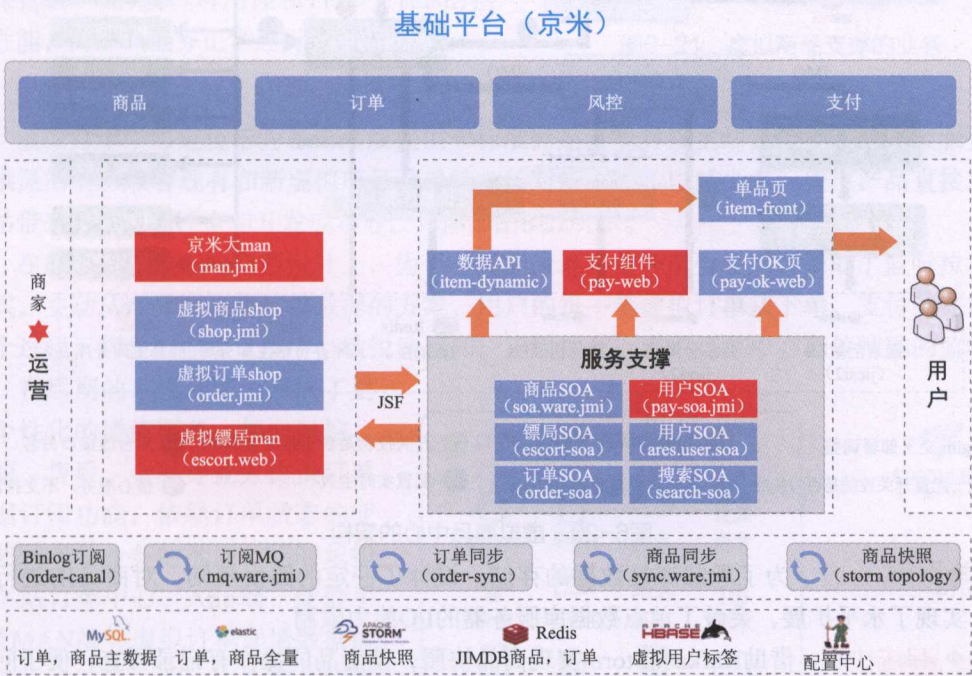


图6-19 虚拟基础平台整体架构

在MySQL、JIESI、Storm、JIMDB、HBase等基础技术的支撑下，不仅抽象出虚拟平台公共的服务模块，如Binlog订阅、商品MQ、商品同步等，还为各虚拟业务提供服务支撑，



核心服务如商品服务、订单服务、风险控制服务、搜索服务、用户服务、数据服务等，核心组件如支付组件、支付OK页、退款组件、结算组件等，力求打造虚拟业务的完整交易流程，实现虚拟新业务快速接入。

1.虚拟商品中心

虚拟商品中心的特征和优势。从各个虚拟业务的实际角度出发，高度抽象商品属性，剥离出商品模型、商家模型、搜索模型、单品页模板一键配置等通用功能，以灵活可配的方式实现各业务商品在发布、编辑方面的管理，并扩展到单品页展示和商品搜索这两个方面。

虚拟商品中心的架构。整个虚拟商品中心由后台配置、商品页面管理、单品页服务、单品页展示、商品消息服务、库存价格服务、垂直搜索及商品中心服务等模块组成。架构如图6-20所示。

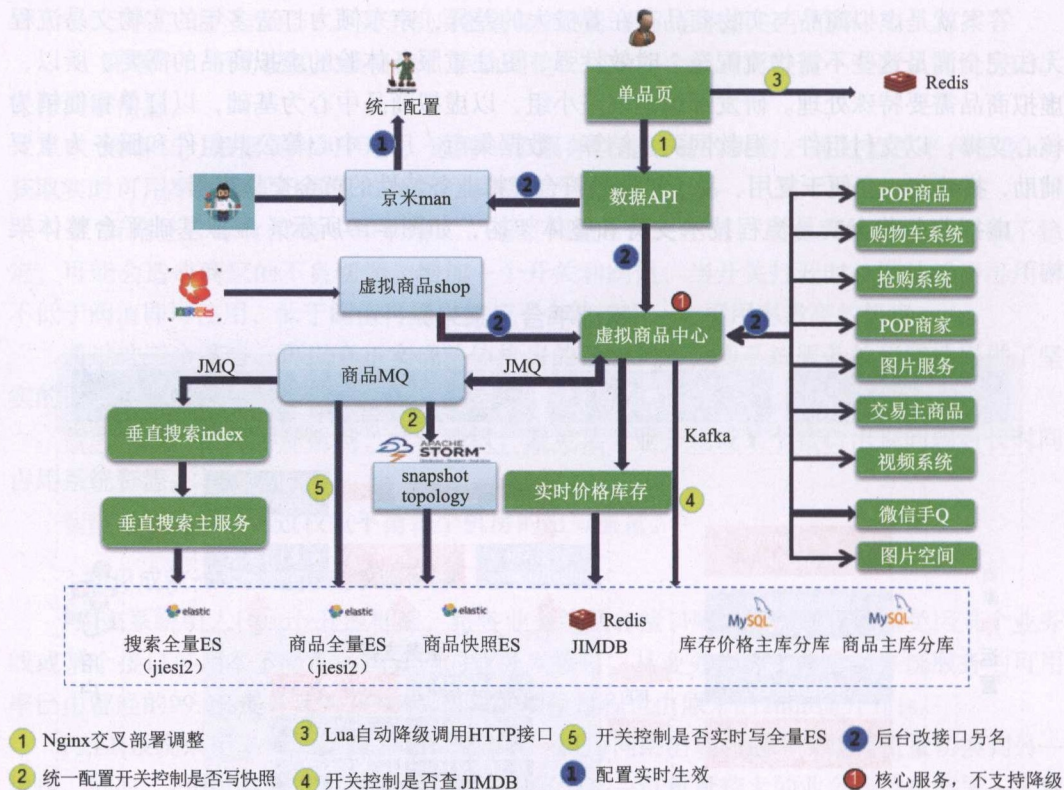


图6-20 虚拟商品中心的架构

◆ **分库分表。**为了承载海量数据的存储，设计了一定的路由规则，对商品ID进行切分，实现了水平扩展，突破了单点数据库服务器的I/O能力限制。

◆ **日志快照。**借助Kafka和Storm实现商品快照，使商品的操作有记录可查，便于问题追踪和数据分析。

◆ **响应速度。**借助Lua实现单品页动态服务，提高了系统的响应速度，保证用户在618期间浏览单品页的顺畅体验。

◆ **数据一致性。**通过MQ消息服务，保证MySQL和JIESI两个数据源数据的最终一致性。

◆ **服务降级。**618期间必须保证业务流程运行正常。为了防止异常情况对交易主流程造成影响，我们在架构上做了各种降级点的设置，一旦出现异常情况，立即启动降级预案。

虚拟商品中心的发展状况和价值所在。虚拟商品中心主要是实现化异为同，为所有虚拟业务提供商品的公共服务。目前已有超过30种业务商品对接虚拟商品中心服务，实现商品发布、编辑及上下架等管理操作，大批量的可通过API接口实现管理，少量的可通过商品管理页面实现管理。目前管理商家的商品SKU数量达到千万级别。图6-21为已接入业务的文字云。

虚拟商品中心在整个虚拟黄金交易流程中处于核心地位，是数据的来源。虚拟商品中心不仅实现了对不同虚拟业务线通用化和个性化的组合支持，也降低了业务人员运营管理的成本。618期间，在流量急剧增长、千万级商品查询的高压之下，虚拟商品的查询在JIMDB、JIESI、MySQL三重数据结构的设计下保持100%的接口可用性和TP99为1ms的接口性能，为虚拟业务正常运行保驾护航。



图6-21 虚拟商品支撑的业务

2. 虚拟订单中心

虚拟订单中心定位为各产品线提供非标准商品订单对接主站订单中心的基础服务，提供灵活的、兼容现有和新虚拟产品订单特性的对接方案，以减少多元虚拟产品直接对接主站带来的沟通、评审和开发成本等，具体如图6-22所示。

在虚拟订单中心的原型设计上，为了尽可能少地对业务产生侵入，采取了监听拉取的方式，主动从产品线收集订单数据的方案，用户的每一笔虚拟订单从下单、支付到完成的整个过程，都会被虚拟订单中心敏锐地感知。正因如此，虚拟订单中心高效精准的监听能力，在后期的系统延伸上开展了更多个性化的增值服务，如向风控、营销、售后、金融等业务提供订单数据订阅功能，依赖订单状态的变更驱动产品业务的推进，同时承担了主站订单中心、App端、微信手Q及MAN端等虚拟订单详情数据展示模板的渲染功能，以分摊各端压力，提供一致的用户体验。

◆ **虚拟业务多维订单系统。**虚拟订单中心的核心功能主要围绕数



图6-22 虚拟订单中心



据搬运工（Hamal）产品运行。Hamal是京米依托开源项目研发的MySQL数据库Binlog监听产品，能够通过伪装slave身份，模拟MySQL的主从复制过程，获取目标库的Binlog事件，并解析出订单数据，高效、高可用、高指标地实现订单数据监听转换过程。

◆ **虚拟订单中心的未来规划。**一个好的服务平台应该是简单的、可扩展的、可维护的、高度稳定的。未来的虚拟订单中心在核心服务上更加专注于订单的汇聚和流转，削减分支环节，不断提升系统的效率和稳定性。同时，对数据监听和订单订阅两大特色服务分别进行产品化和服务化推广，前者的无侵入监听转MQ的设计是值得借鉴和推广的异步驱动模式，后者订单的变更订阅已经开始为京东内部多个业务场景提供订单后增值服务。随着京东海外业务的不断拓展，居家旅行生活服务类的虚拟产品有着快速融入和推广的巨大优势，这也对虚拟订单中心支持国际化发展提出了新的需求，未来需要在数字货币转换、多语言支持、国家特色化、跨地区部署等方面进行改造。最后，虚拟订单中心运维以来的沉淀与积累的海量订单数据是十分宝贵的资源，如何通过大数据处理和挖掘进一步发挥这座金矿的价值，决定了虚拟订单中心未来的发展方向。目前虚拟订单中心从用户角度出发，着手通过订单数据分析构建用户虚拟画像，以为虚拟产品提供千人千面的精准化营销，同时也将关注虚拟产品之间的联动性，积极开拓订单数据在风控、促销、推广等方面的挖掘与探索。

6.3.3 京东户簿建设之路

随着业务的不断发展壮大，越来越多的业务系统需要对用户进行身份信息的实名认证。该流程会涉及用户身份信息、图像信息的上传、加密、存储、认证等相关工作。这些信息在多个业务系统中管理和维护：一方面，认证标准、流程、操作界面不统一；另一方面，无法保证数据的安全性、一致性和复用性，对这些敏感数据请求的监控成本越来越高。为了解决这一问题，需要建立一个具备高安全、高性能、高可用、人工智能、通用性好的身份信息统一管理平台，京东户簿由此诞生。

1. 稳定的运营

京东户簿经历了信息归集、各业务系统的历史数据同步与实时数据同步、上传组件接入等阶段，现已正式上线，为各业务系统提供服务，已经接入的业务线包括全球购、手机运营商、E卡、节能补贴、飞机票、火车票、山姆会员、宽带运营商、海外房产、加油卡等，成功地满足了各个业务线的需求，系统运行平稳，目前证件信息的数量达到亿级，单日请求量数百万次。

2. 健壮的架构设计

从整体架构来看，户簿系统由几个子系统构成，系统间协同工作。组件上传系统为业务提供公共组件上传服务。中心系统面向内部系统和业务系统提供完整的、公共的服务。信息同步系统应用于历史数据同步的场景和数据补偿场景，具体如图6-23所示。户簿管理端可用于业务系统接入申请、权限、密钥申请等管理工作，各系统户簿采用CQRS的模式，在避开资源竞争、网络开销（IO）的同时提高了查询性能。CQ两端代码层分离，C端对数

数据库进行写入，Q端对ES进行检索完成复杂的业务，同步数据采用异步方式，实现数据的最终一致性。

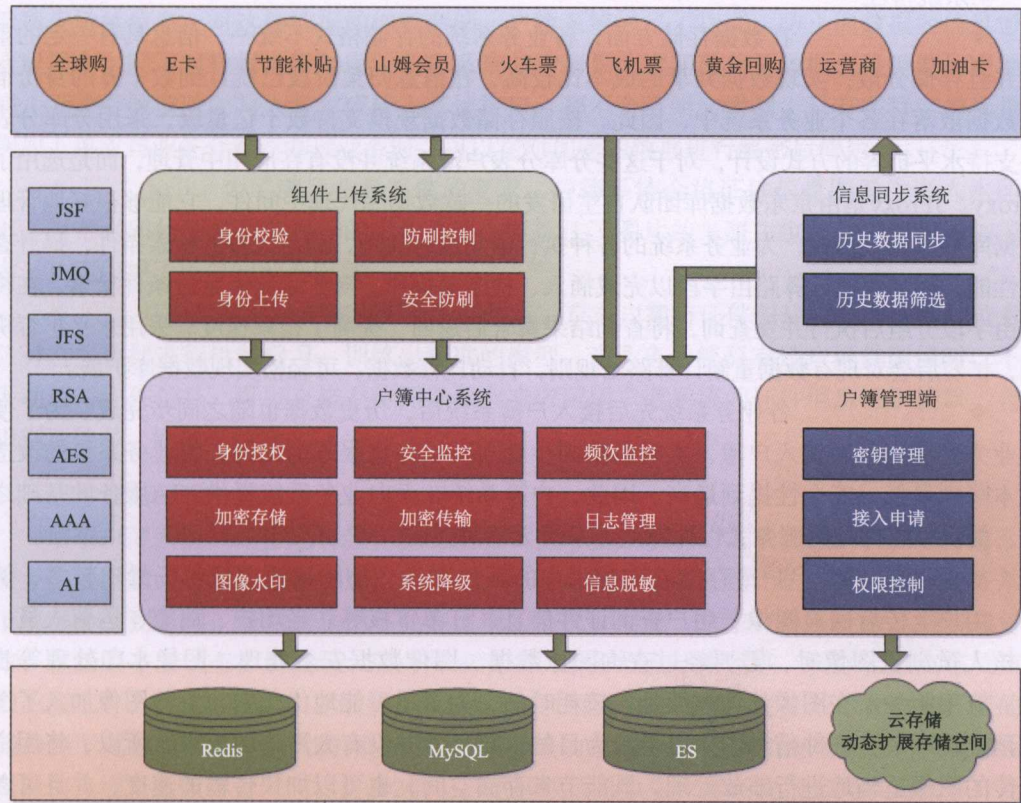


图6-23 户簿系统的架构

◆ **系统降级服务。**当系统负载过重时，在水平扩容节点之外可对图像上传容量进行降级设置，动态降低图像上传的容量限制，以缓解服务器承载压力，配置校验将立刻生效。

◆ **可用性保障。**缓存在满足高性能的系统中扮演着重要的角色，缓存的读写速度远远超过数据库，成为众多应用在提高查询速度时最为常用的一种方式。当然，过分地依赖缓存、滥用缓存的后果也是非常严重的。为了提高系统健壮性，户簿必须进行缓存宕机演练，在缓存系统出现问题并宕机的情况下，使用缓存的环节可由数据库或其他存储引擎代替工作，会话信息可在请求中携带，通过通信解密、解析等方法完成会话复原，系统依然可以正常运行，使系统满足最低可用性。

3.安全与高效的应用部署

户簿系统采用分布式部署方式，内网、公网系统均有自身的安全及预警机制，户簿组件对外提供服务，在安全防刷、身份校验、权限控制等方面做了充分的考虑和设计，采用集群部署动态切换流量，中心系统服务位于内网部署，是户簿的核心系统，可支持动态扩充服务节点，轻松保障系统性能，中心系统与各服务之间的通信均采用加密传输、校验签名机制保障数据的安全，信息安全系统也对户簿进行实时的安全防护，提供实时校验、拦截的服务。



关于存储，数据库、缓存云、ES均采用集群部署、一主多从，保证了数据的可靠性。

4. 系统特性

◆ **存储可扩展。**在数据存储方面，各业务系统的存储格式不统一，信息具有一定的重复并且存储分散，实现数据共享的成本比较高，在信息归集阶段已统计到数千万的身份信息数据散落在各个业务系统中，因此，按照存储数据规模支持数十亿量级，采用分库分表并支持水平扩容的方式设计，对于这些分库分表户簿系统并没有在应用中管理，而是选用了JProxy。JProxy是由京东数据库团队自主研发的一款数据库代理中间件，它能够很好地管理数据库分库分表信息，为业务系统的各种执行语句进行代理，满足大量的检索条件，提升查询性能，例如快速计算路由字段以完成插入、检索等操作。对于多路由字段条件检索，在将路由字段分组后执行并发查询，将查询结果聚合后返回，提高了检索速度。采用水平扩容机制，扩容后会对现有数据重新计算路由规则，自动迁移数据，可轻松实现数据库扩展。

◆ **组件化服务。**各业务系统先后接入户簿系统后，历史数据也随之同步完成。为了使各业务系统完整地接入户簿，实时数据同步就需要组件化服务的支持，将业务系统的改造成本降到最低，安全性提到最高。因此，户簿系统在面向业务系统提供SOA服务的基础上，提供了组件化的服务，以保障业务系统无缝对接。

◆ **图像预处理。**户簿存储着大量用户的证件图像，访问这些图像的场景比较多，例如，用户上传后回显图像、用户查询证件信息、订单审核等。当用户、商家、运营人员、审核人员浏览图像时，需要经过查询图像数据、图像数据安全处理、图像水印处理等步骤。应用服务器在图像处理方面是比较耗时的，为了尽可能地优化它，上传图像加入了图像预处理环节，证件信息图像以看清为目的，对尺寸并没有太严格的要求，所以，将用户上传的图像适当地进行缩放处理，既能节省存储空间，也可以加快传输的速度，并且可以提升后续处理的性能。缩放图像之后，开启一个图像处理任务，将水印文本融合到图像中进行存储。这样，在接收图像请求时，只需要将带水印的图像返回调用方即可，提升了获取图像的性能。在实际情况下，水印文本不会被轻易地更换，因此带水印的图像也不会经常重构，实现了空间换时间的思路。

◆ **多维度查询。**为了满足业务系统多维度查询，数据库无法满足查询的性能要求，户簿系统使用搜索引擎来满足业务查询需求。在查询对写入比例较高的情况下，采用CQRS架构模式是一种不错的选择，数据库负责数据的写入，搜索引擎负责数据的检索，不仅提高了检索的性能，也为写入节省出大量的资源。关于数据同步方面，采用顺序队列消息机制，将数据库变化的消息发送到同步队列中并且携带版本号，在消费此消息时校验版本号，成功后进行更新，最终保障数据更新的一致性。

◆ **安全访问控制。**从安全访问的角度考虑，各业务系统对访问者的身份校验、权限控制、日志记录、请求监控、预警、阻断等方式各不相同。户簿系统为了解决这些问题，将这些安全点设计成标准的、统一的处理方式。例如，接入户簿的业务系统都需要进行接入申请、授权、创建权限等步骤，当访问者提交请求时，系统会校验访问者的身份及权限并记录相关的操作日志，除了户簿系统本身校验之外，信息安全系统也会对请求的地址、应用进行校验，对于未授权的应用进行阻断并发出警告，防止数据泄露。

◆ **安全存储与安全传输。**从数据的安全存储、安全传输等环节考虑，各业务系统处理方式各不相同，例如加密方式、加密算法及密钥本身的安全处理不同，证件号码和证件图像属于敏感信息。对这些数据的加密存储，户簿系统做足了功课，防止因数据库信息泄漏造成损失。除了对数据进行加密，业务系统与户簿系统之间通信的数据也是需要加密处理的，无论是被监听、写入日志或传入其他接口，都不会将明文泄漏。只有在特定的场景，例如下单或与业务合作商办理业务时，可以在对加密的数据进行解密后使用。

◆ **图片水印。**为了满足业务系统的要求，户簿系统会将证件信息图像中嵌入水印文本，附带水印的图像具备很好的干扰效果，既标注了图片的来源，也为盗用图像增加了成本，从而保护了用户的私人信息。

◆ **智能识别。**人工智能已成为当下的热点，京东户簿将证件图片数据提供给AI技术团队对这些数据进行机器学习，训练出精准的模型，当业务系统只有用户证件图像时，不必手动填写证件信息，可以直接上传图像保存或者调用户簿系统的证件信息识别接口，自动识别证件号、姓名等信息。

5. 性能指标和价值体现

京东户簿系统从诞生之日起，就已经融入各个业务系统的下单流程中，如果户簿系统服务出现问题，将意味着整个虚拟、实物添加身份信息下单流程无法继续，很多业务系统下单都会受到影响，显然这是绝不允许出现的情况。因此，京东户簿系统做了很多准备工作，包括服务器动态扩展演练、服务降级演练、服务节点分流演练、缓存宕机演练及消息积压演练等，在这些演练中不断优化系统、提高性能。在618期间的实战中，户簿系统的性能得到了充分的认可，全天访问上亿次，服务成功率100%，处理消息峰值每秒5000次/s，身份信息增量达数百万（同比提升2倍）。这些成绩是京东户簿系统很多同事无数个日夜研发与优化、辛勤付出，以及多部门的支持与配合的结果。京东户簿在未来的建设道路上还需要不断地自我完善与发展，服务更多的业务系统，做出更大的贡献。

6.3.4 虚拟业务的风控挑战

虚拟商品相比于实物商品最明显的特点是交易周期短，一手交钱、一手交货的买卖通过互联网可能在几秒内完成，例如话费、加油卡、游戏等充值类业务基本都是实时到账。交易周期、流程和中间环节越少，把控风险的挑战就越大。部分产品，例如电影票、机票等，存在支付前有限资源的占用，且转化率和支付率常常较低。与此同时，京东的很多虚拟产品都处于快速成长的市场拓展阶段，严重依赖促销来培养用户黏性和吸引新用户。促销方式的多样化、大力度、高频率等特点极易引来刷单风险或其他恶意行为。京东每年的618是吸引新用户和流量提升的关键时期，恶意刷单等欺诈行为严重影响用户购物体验，造成促销资产的浪费，虚拟镖局系统临危受命，肩负起维护虚拟平台交易的正常秩序、保证用户体验的重要使命。

1. 哪些场景下需要勒紧风控

虚拟产品短暂交易期内的风控判定有多种形式：一方面可以根据用户特征数据正向评



估，对用户的登录方式、购买习惯、下单转化率、退款率、是否实名等进行标记评分；另一方面可以从用户历史下单行为出发逆向分析，对本次购物流程中的领券、预占资源、用券下单、退换货等行为进行检测拦截。相关的用户因子包括：用户等级、注册时间、是否实名、购买力、实物风险等级、金融风险等；相关的行为因子包括：是否使用优惠券、浏览行为、购买总数、购买历史、支付转化率等。

虚拟产品交易环节中需要控制风险的场景如图6-24所示。首先，在用户下单前，有验证是否实名认证、是否超买、是否有不良行为记录等控制机制。同时，对于需要提前占位销售的紧缺资源，如电影票、演出票、机票等，下单前核查用户等级、历史信用、风险等级，避免恶意占位等情况。在账单结算环节，有检测用户IP定位信息、安全支付方式、用户PIN异常信息等控制机制，避免账号盗用下单。此外，批量



图6-24 风控场景

注册会员，伪造新用户骗取优惠大礼包是最常见的欺诈。新用户购物的随机性大，也没有任何的购买行为记录，计算风险的可用信息极少，普通的拦截规则大部分都会失效，贸然拦截又会有大概率的误伤，因此必须挖掘历史新用户的注册环境、浏览环节、购买意愿、支付结算等数据进行综合评估，有效地进行新用户首单全流程风险控制。

2. 镖局实时计算风控中心

虚拟镖局实时计算中心是京米平台完全自主研发的风控拦截系统，从立项到上线运行，可谓一波多折。由于风控的拦截必须要渗透到产品的业务逻辑中，且流程设计和规则模型计算都极其复杂，经过大量的项目调研、方案讨论和产品推广，如今虚拟镖局已经得到越来越多的产品线的支持和认可，镖局的风控接口已经嵌入虚拟用户购物流程的多个环节中，默默地为虚拟业务提供金盾保护。

风险规则的计算依赖大量的源数据（订单数据、用户数据、商品资源、财务资产、信息安全等），同时对实时性能指标要求极高，是典型的请求—响应模型。受常见高并发Web服务容器的启发，我们设计了主要包含请求路由、拦截过滤、会话管理的容器管理方案，在session scope中存储用户模型数据，在request scope中存储用户行为模型数据。类似Web应用，规则的计算流程体现为容器内的请求处理流程。

镖局容器启动后，在准备阶段完成所有组合规则的规则栈初始化工作，根据风控规则的优先级进行排序、过滤、去重等优化，优先计算线上规则，延迟异步计算灰度规则，风控规则栈表达为模板形式，通过模板计算结果映射为风控建议（此处理类似HTTP视图和数据绑定计算过程）。

虚拟镖局实时计算中心在设计上分层清晰明确；虽然引入了大量高并发任务设计，但请求处理和规则计算上相互隔离，接口功能单一；对外兼容JSF和RestFul等多种请求方式，业务接入简单快捷，服务运行稳定高效，正逐渐成为虚拟产品业务发展的强力守护者。

3.旗帜的力量

每年6月1日开启购物狂欢模式，京东的虚拟业务（如生活服务、旅游度假、游戏充值等产品类目）在不同时段开启了优惠券发放和一波高过一波的整点促销活动。虚拟镖局也为机票、游戏点卡、加油卡等产品在领取优惠券、购买促销商品、下单流程等多个环节和场景新增了近200条风控规则，以保证促销的正常进行。在整点大促期间，虚拟镖局的旗帜累计提供了百万余条风控拦截建议，累计过滤订单金额超过数十亿元，线上识别恶意用户和恶意IP达数十万个，拦截了大量重复使用优惠券恶意购买充值类产品的刷单行为、异常登录，以及恶意占位后不支付或取消订单等高风险行为。在服务性能方面，风控计算中心给出了TP999维持在接近80ms，平均耗时14ms的好成绩，有力维护了618大促期间虚拟产品的正常秩序和用户体验。

6.3.5 虚拟业务全渠道推进

随着虚拟全渠道推进的展开，虚拟业务从起初单一的Web应用到移动App、H5、微信，再到ONEBOX、商旅、大客户等多种渠道。2017年印尼机票这一产品，开启了虚拟业务走向国际的新篇章，打响了虚拟产品开疆扩土新模式的第一枪，相应地，新扩展的多种渠道也会在新市场根据需求一一推进。京东印尼站是印尼当地的综合性电商平台，旅行市场在印尼是仅次于零售市场的第二大市场，因此，京东印尼站搭建机票频道是战略拓展中很重要的一环，Traveloka是印尼排名第一的OTA，与Traveloka的合作可以为印尼站快速进入市场提供便利。在本次合作中，Traveloka通过API的形式提供产品，并对印尼站机票的售后负责，高度复用和快速持续交付是我们快速将印尼市场提供的系统服务于新国家市场的有力保障，并且微服务化能够保障快速持续交付，还能根据市场需求将微服务快速组装成新型功能，以满足新兴市场的需求，从而保障快速占领市场的战略需要。

1.微服务化征程：充实且丰富的虚拟军火库

在微服务架构系统中，每个服务都是独立的业务单元，职责单一、代码库独立、服务进程隔离，当对某个服务进行改变时，对其他的服务不会产生影响。这样，每个服务都有独立的测试机制，无须考虑破坏其他功能而建立大范围的回归测试，每个服务都在各自的进程中独立，因此，从部署的角度考虑，服务与服务之间也是高度解耦的，这样能保障持续快速交付。图6-25展示了采用微服务架构，并发开发时各阶段工作是高解耦的。

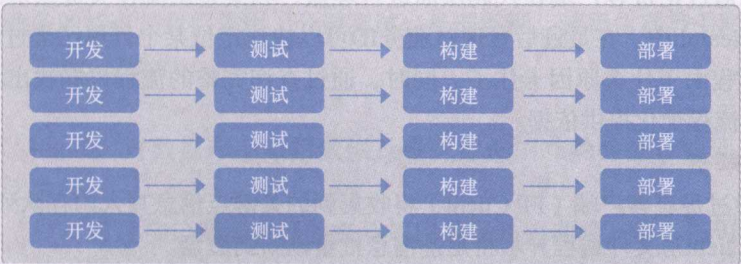


图6-25 微服务架构并发开发各阶段工作高度解耦

随着微服务化征程的深入，虚拟军火库里的装备越来越齐全，如虚拟基础服务组输出



了支付、退款、结算、镖局、促销、订单中心等内部公共服务，一方面减少了虚拟内部各业务线重复建设问题，另一方面降低了因依赖服务的升级虚拟各业务线的升级成本。图6-26展示了当前虚拟提供的部分基础服务。

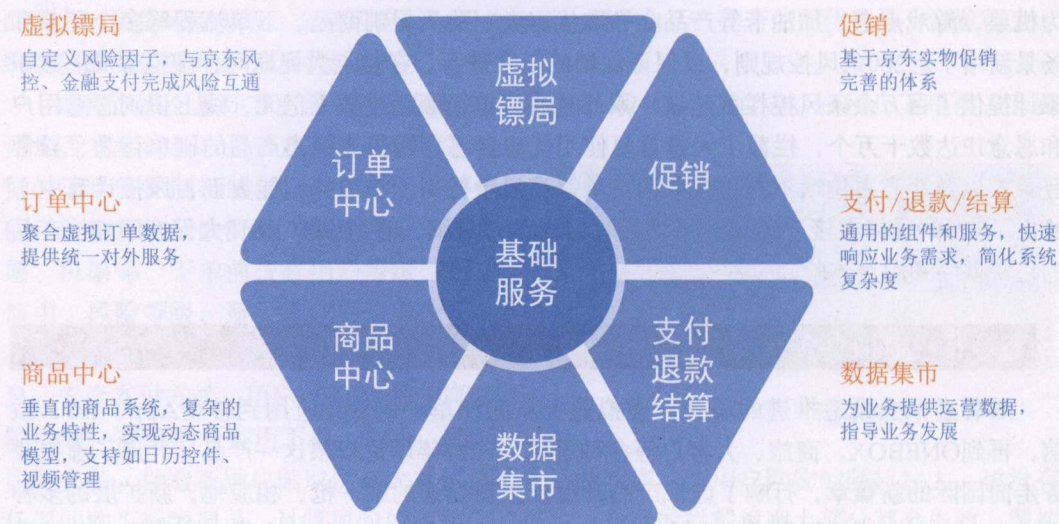


图6-26 虚拟当前提供的部分基础服务

2. 足迹系统：虚拟世界问题的终结者

在每年的618期间，会产生海量的订单，那是虚拟平台业务线日夜备战的日子。在高速发展的虚拟业务线的备战中，需要为618期间的线上系统做好各种降级措施，并且能快速地解决线上的各种突发事件。在庞大的分布式系统中存在着错综复杂的调用关系和外部系统依赖，这个庞大的分布式系统对于如何快速定位问题并以最快速度的解决问题提出了很大的挑战。

例如，火车票系统购票流程比较长，分为支付、预定占座、出票、退票、退款。为了系统的性能，又将每个步骤分为同步调用和异步通知，流程长度相当于增加了2倍。在很多实际业务场景中，都需要快速准确地定位某个订单当前走到了哪个流程或哪个流程因为什么原因卡住了，如果通过分布式日志系统查找会比较慢，而且有些时间比较旧的日志会被删除，就更无从查找了，因此，单机部署遇到分布式部署产生了分布式日志收集系统，单体应用遇到微服务产生了分布式调用跟踪系统，短流程遇到长流程产生了足迹系统。

足迹系统通过足迹客户端在代码层面进行埋点，收集系统流程关键点信息（足迹）并保存在ElasticSearch中。在足迹后台通过业务ID可以快速定位某个订单已走过的流程，以及在某个具体流程由于什么原因卡住了。同时，通过查询完整的流程也能找出系统中不合理的点，从而为流程优化提供依据。

足迹系统解密

- ◆ 应用通过足迹客户端（FP Client）将收集到的关键点信息写入内存队列。
- ◆ 由FP Client写文件线程异步将足迹信息批量刷新到磁盘。
- ◆ FP Client再通过低优先级线程将磁盘上的足迹发送到远程FP Server，将其持久化到ElasticSearch中。

◆ 通过根据业务ID在足迹后台系统可以快速查询该业务经过的完整流程结点。

FP足迹查询平台。通过业务ID可以查出对应的足迹内容（内容）、足迹发生的时间（足迹时间），能快速知道当前业务流程走到了哪一个环节、处于什么状态，在大促期间能让售后、运营、客服更好地为用户提供服务。

由于虚拟业务系统大量使用MQ消息来驱动业务异步处理，并且火车票业务会带来大量抢票订单，因此在618备战期间，当使用FP足迹系统快速查询定位到线上问题时，经常会修改消息处理的核心业务代码并在预发布环境中测试。在快速迭代上线过程中，为了保证预发环境的测试不会影响线上消息的处理，让预发环境只处理内部测试时产生的消息，线上只处理线上用户产生的消息，达到预发和线上消息隔离的目的，产生了消息的预发灰度。

火车票的消息预发灰度通过在管理后台配置测试pin，当接收到消息时基于pin来识别消息的来源，判断该消息是测试的消息还是线上的消息，进而决定是直接消费还是需要转发到目标消息队列topic再由目标环境consumer进行处理。

通过足迹系统能快速定位线上问题。当问题修复后，通过消息的预发灰度能安全避免快速测试上线带来的系统风险，既保证了系统的健壮性，也能更好地提升用户体验。

3.全渠道拓展：开启虚拟开疆扩土新模式

到目前为止，虚拟产品主要分为旅行和生活两大类，已上线的产品有国内机票、国际机票、国内酒店、国际酒店、手机充值、火车票等29个产品；国外市场也在迅速发展，即将上线的有印尼机票、印尼充值，以及规划中的印尼酒店。根据当前的用户习惯，所有服务都会支持PC版本、移动App版本和H5版本，保障用户在京东的一站式体验。

6.3.6 质量保障与实战方案

商家众多。如何安全接入？如何实施执行万亿量级的测试案例？怎样才能提升项目整体测试质量？本节将从测试方法、测试工具、质量评估、风险预测四个方面解答上述问题。这些手段筑起的长城，能够全方位保障虚拟平台618的顺利运转。

1.方法先行：快速接入与质量保证

不同于其他业务，虚拟业务具备许多特殊的属性，如无实物、无物流、无库存、订单时效性强、相对独立等。其中最突出的就是接入商家数量多、种类多，如何保证不同商家的安全接入就成为了质量保障的关键要素。

针对虚拟业务的特点，我们采用了分层测试、外部解耦、流程仿真等手段以确保每个接入商家的质量，如图6-27所示。

分层测试

◆ **DATA层测试。**对数据库中原始数据及聚合数据的准确性进行验证，如精度、数量、存储有无丢失等。

◆ **API层测试。**通过接口调用验证服务器返回数据是否准确，服务端可能会将数据进行运算后返回，通过接口层验证保证数据传输和服务端处理的准确性。

◆ **UI测试。**通过覆盖系统所有功能保障系统UI交互层的正确性。



外部解耦。系统的运行离不开外部系统的依赖，尤其是商家对接、网络环境、消息依赖、数据依赖等。测试过程中如何降低系统间的耦合度是能否高效进行测试的关键：MQ消息自动发送组件模拟商家消息传输，可以解耦系统间的消息依赖，满足快速测试、高频迭代的要求，如利用该工具模拟机票业务出退票消息，可成功摆脱消息依赖，使测试效率及准确性大大提升。

流程仿真。商家接入在一些极端情况下很难通过流程模拟，这会造成测试覆盖不全的情况，依据穿线测试理论，可将流程主信息进行标记追踪，根据不同商家将流程引导至我们设定的极端情况中，验证系统处理能力。

当然，仅仅保证功能的正确性及可用性还远远构不成强壮的系统，为了保护用户及公司的利益不被侵害，健康的系统还应具备风险应对策略及处理能力，在这方面测试人员又需要如何应对呢？

2. 高效之路：测试工具与实践效果

在系统风险处理策略中，任何合理的测试设计都不应被视为过度设计，全覆盖的测试才是系统检测的最优选项。

例如，虚拟镖局系统测试，覆盖所有风险因子组合的测试用例有7.938万亿条。面对这近8万亿条的测试用例，人工执行显然是不可能完成的任务，通过自动化来解决也不能很快得出结果。在时间紧、任务重的618备战时期，这无疑是一个必须解决的问题。

为了解决这个难题，我们实现了正交用例生成工具（风险模型的生成）、关系型数据上中下取值用例生成算法（用户下单数据的生成）等，在不降低测试覆盖度的前提下，减少了测试用例的空间复杂度，最终通过使用上述两款工具将近8万亿条测试用例减少到32902272条。

通过多线程并发调用接口，历时5h完成全部测试用例的执行，并发现手工测试难以覆盖的多个严重Bug。软件系统的业务测试除了需要对系统逻辑的合规性、系统交互的易用性进行验证之外，还要保证系统数据的正确性，我们又是如何做到的呢？

3. 通用为本：全量数据比对与分析

虚拟平台业务的独特性，决定了被测逻辑的复杂多样性。由于在交易构成中，订单状态的转换非常快，在海量订单数据中寻找异常订单数据如同大海捞针，靠人工反复通过SQL检查多个数据源的订单异常状态就成了不可能完成的任务。

通过订单异常监控工具可以完美解决以上痛点。虚拟订单需要在多个数据源中进行存储，因此校验订单数据一致性成为测试的重中之重。我们开发了订单状态异常监控工具，通过数据库快照的形式保留订单的过程态，让本来在业务流程中转瞬即逝过程态保存在快

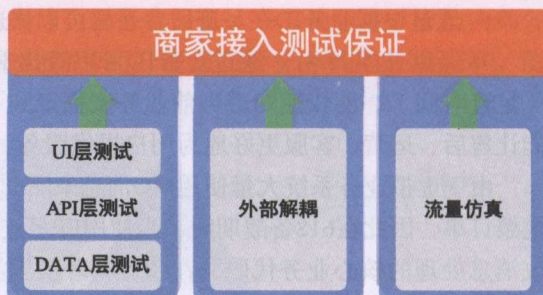


图6-27 商家接入测试保证

照中，实现了订单数据一致性校验。

该工具源于虚拟平台，适用于整个质量保障行业。在助力618的同时，不仅在质量和效率提升方面起到了关键作用，且在质量评估和风险预测中也得到了印证。

4.测试上移：质量评估与风险预测

在618备战期间，面对众多需求，测试工程师们都做好了充分的准备，从测试用例的撰写、测试数据的准备到测试执行过程，都进行了严谨的评估、评审。除了常规的系统质量保障和评估方法，我们开发了送测项目分析工具，通过缺陷管理系统的REST接口获取虚拟平台的全部测试项目信息，快速查看项目状态变化，掌握项目推进速度。对缺陷管理系统的信息研究过程中，发现一个项目的缺陷解决周期和项目风险存在内在联系。

在一个项目里有多个缺陷，每个缺陷都有一个解决时间（是一个时间间隔），设 $P=(Tf_1, Tf_2, \dots, Tf_n)$ 。通过一些处理计算出每个项目的样本标准差、样本均值和样本90分位数。这样每个项目均可用 $P=(dev, eva, per)$ 转换公式表示。然后，将缺陷管理系统的全部项目都做以上计算，得到一个 $N \times 3$ 的数据集合。这个集合看似无用，但通过K-Means算法计算，奇迹出现了，如图6-28所示。

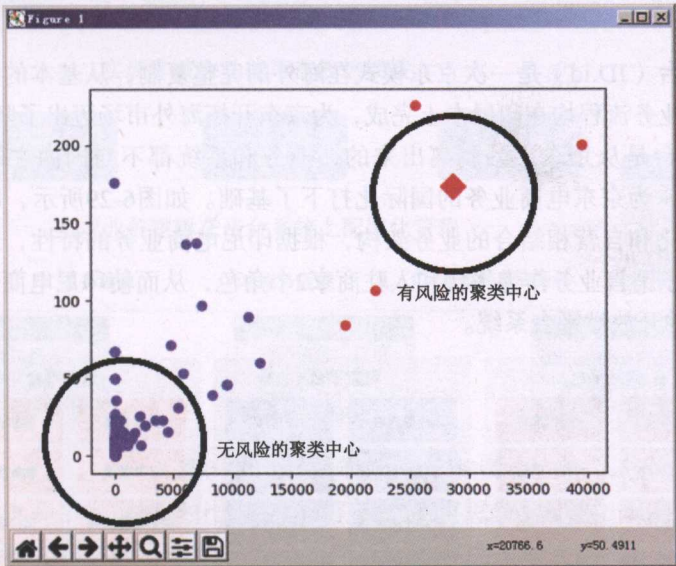


图6-28 K-Means处理的缺陷特征集合

蓝色区域的聚类中心就是无风险的项目，红色部分就是发生过风险的项目。在618备战期间，将对正在进行的项目转换数据，发现欧几里距离越接近蓝色聚类中心越不容易发生风险，相反则越容易发生风险。

当一个项目上线后，对应项目又会进入训练集合纠正聚类中心的位置。通过这样的不断自纠错，该算法能越来越早地发现项目风险，保证项目的按时交付。

在618期间，虚拟质量组采取了一系列措施和手段保证系统的稳定运行，出色地支撑了虚拟平台百万级的订单量，虚拟系统稳定如常。



虚谷：虚拟研发部架构师，负责虚拟黄金交易流程的技术架构。
znmars：虚拟平台架构师，负责应用平台研发组项目的系统架构工作，并多次负责618备战的组织工作。

6.4 创新业务

6.4.1 京东的“海外丝绸之路”

未来3至5年是中国电商走向世界的大发展时期。国际化是京东五大战略之一，京东将致力于满足全球用户的需求，向全球消费者输出“中国好品牌”和“中国好商品”。京东海外电商平台研发部通过技术驱动电商交易，提升效率，致力于让海量“中国好商家、中国好品牌、好商品”乘着京东的大船走向全球，通过国际伙伴战略合作关系及海外本地化运营，给予全球买家更好的用户体验，并且提供公平健康的生态环境，保障商家在京东海外平台长期、健康地发展，做全球最值得信赖的电商平台。

1. 扬帆印尼

印尼电商平台（JD.id）是一次京东模式在海外的完整复制，从基本的电商流程到供应链、物流配送等业务流程均在印尼本土完成，为京东开拓海外市场迈出了坚实的一步。

印尼电商平台是从京东主站剥离出来的，业务和系统都不与国内主站耦合，独立运营，可快速复制，为京东电商业务的国际化打下了基础。如图6-29所示，印尼电商平台在架构上采用平台化和自营相结合的业务架构，根据印尼电商业务的特性，系统设计之初便将商城运营和京东销售业务作为平台和入驻商家2个角色，从而使印尼电商平台成为可容纳自营和POP入驻的开放性平台系统。

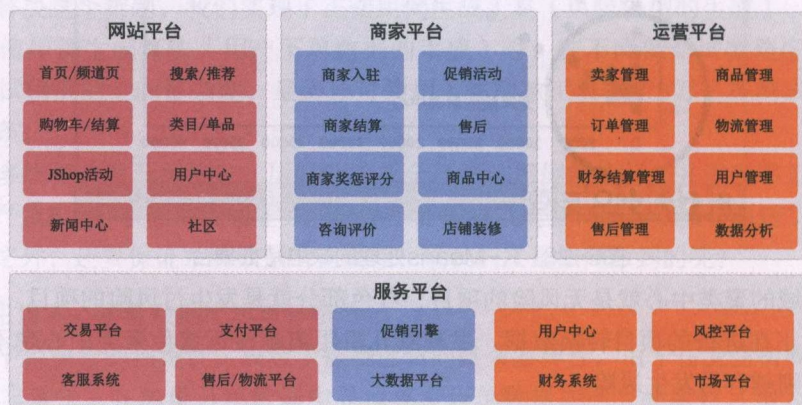


图6-29 印尼电商平台业务系统图

整个印尼电商平台由网站平台、商家平台、运营平台、服务平台几个核心的模块组成，实现了客户在PC端、移动端集购物、支付、客服、售后、配送等全购物流程及后台采购运营管理的全流程。

整个印尼电商平台特点：

- ◆ 独立于京东现有系统：印尼电商平台拥有独立的商品、促销、运费、售后、支付、搜索、广告投放点击等系统；
- ◆ 平台微服务化：业务高度聚合，具备多语言、可复制、可扩展的产品化能力；
- ◆ 耦合度低，易于扩展：由34个基础子系统、32个接口系统组成，支付系统对接当地20多家银行系统；
- ◆ 成功对接Facebook、Paypal、Google、Line等比较成熟的海外平台。

2. 技术积累

如图6-30所示，印尼电商平台的基础系统已经进行了平台化改造，商品中心、用户中心、交易中心等都完成了平台化的升级，基础服务下沉让系统核心更加稳定，即使面临618这样的大促考验，也可以有针对性地保障基础平台的稳定性和高性能。

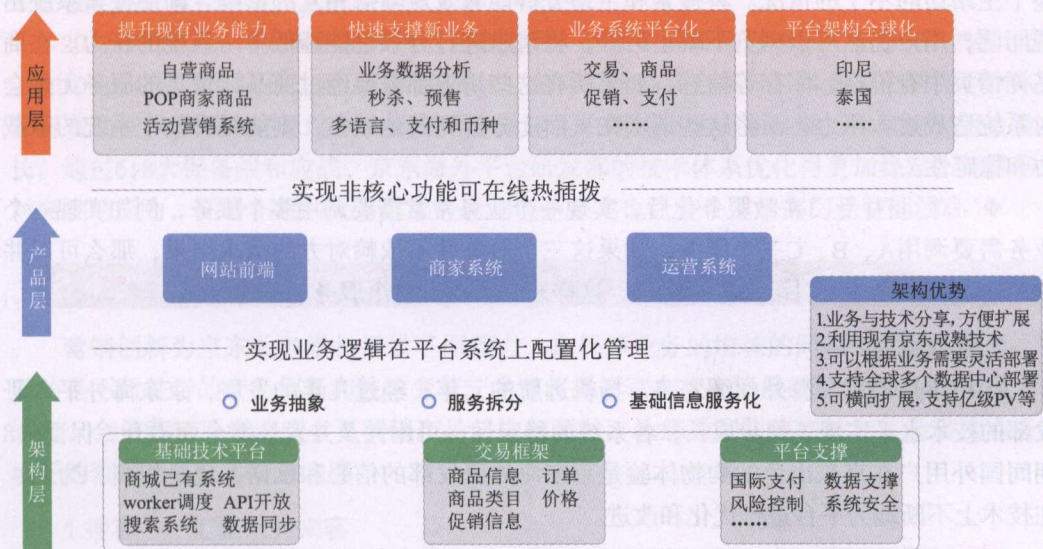


图6-30 印尼电商平台架构图

在整个印尼电商平台升级优化前，提出了两个值得思考的问题：印尼电商平台目前有哪些瓶颈直接影响和导致线上的稳定性，如何从根本解决这些问题？如何既满足业务增长的需求，又保证系统稳定？

通过以下策略来优化整个印尼电商平台：

- ◆ 异步化。异步化能很好地控制并发，并提高效率。例如在用户注册的过程中，如果涉及日志的记录，将日志记录做异步化处理能很好地减少注册功能的响应时间。另外，针对一些实时性要求比较高的场景，如库存的扣减，仍然可以用异步化的思想，将扣减在缓存中进行操作，异步到数据库中。这样能减少数据库的并发压力，当然也需要考虑实现缓存和数据库中数据的最终一致性。
- ◆ 多级缓存。缓存在业界各个系统中都用得很多，也被谈及得很多，说明缓存仍然是提升系统性能的一大利器。从浏览器缓存到CDN，再到Nginx缓存、JVM堆缓存、分布式缓



存，从前到后，将用户所需要的信息尽量快地返回给用户。例如，印尼电商平台的M站、PC站首页等实时性要求不是很高的数据，在浏览器和CDN缓存后，大量的请求不用经过后台服务器就可以返回给用户。对实时性要求比较高的数据，如库存数据、促销数据，仍然可以在Nginx中缓存几秒来有效地控制并发请求。除了这些常用的缓存，我们还使用了Google的PWA（Progressive Web App）技术，让用户在移动端断网的情况下仍然可以浏览内容。

◆ **降低依赖。**在大型分布式系统中，如果访问量比较大，有一个系统响应慢，可能会导致整个系统雪崩效应的发生。所以，根据业务的重要性，有一些优先级比较低的服务，针对这个服务的调用超时时间可以设置得低一些。访问超时了就不返回结果，即相当于降级了，例如商品评论。还有一些服务，例如在商详页的库存查询，如果库存访问超时就默认返回1，让用户可以进入结算页下单，在结算页和下单流程中会对库存做校验。

◆ **系统拆分，请求分散。**在印尼电商平台的促销活动中，曾经发生过搜索量过大导致整个主站访问不了的情况。将搜索和主站分拆能有效地降低相互的影响，即使搜索系统出现问题，用户仍然可以浏览下单。另外，将请求进行分散也能降低单一系统的压力。在商品详情页中有价格、库存等信息，如果所有这些请求都需要经过商品详情页的服务无疑会对系统造成过大压力，而将这些请求在页面上分拆到其他域名，则能提升整个系统的承载力和稳定性。

◆ **串行改并行。**在微服务化后，实现一个业务常常需要调用多个服务，例如实现一个业务需要调用A、B、C三个服务，如果这三个服务并不依赖对方的请求结果，那么可以并发调用这三个服务再对结果进行组合，这样能大大提升整个服务的响应时间。

3. 追求卓越

2015年是京东“海外丝绸之路”扬帆远航的元年，经过几年的发展，京东海外平台研发部的技术水平实现了稳步成长，各系统的稳定性、可用性及并发性能全面提升。保证618期间国外用户在京东出色的购物体验是京东海外研发部的信心和承诺，也是担当，为此，在技术上不断地为平台进行优化和改进。

提升性能

◆ **重构接单服务代码，使数据库读写分离，进行异步并行与串行调用相结合，提高接单处理效率，支持百万级单量/天。**

◆ **完成库存下单逻辑异步化改造，TP99整体性能从300ms以上提升到50ms以内，库存整体性能提升3倍以上。**

◆ **商品单品页、销售属性等采用智能排序算法，查询性能提升10倍左右。**

提升平台安全性

◆ **接入京东风控系统，支持用户黑白名单，配置更统一化、精细化。**

◆ **多次对站点进行全面的安全漏洞专项测试及优化。**

◆ **Nginx防刷系统优化，支持IP黑白名单，配置更统一化、精细化。**

◆ **用户注册、登录、支付启用强制HTTPS，确保用户信息安全。**

提升平台稳定性

◆ **每次618大促前，进行数次的线上压力测试，有效评估了购物流程关键节点上各个**

系统的承压能力，同时锻炼了团队快速定位问题，评估影响和解决问题的能力。

◆ 通过专人专项快速梳理，整理出了6个维度、25个细项的大促备战任务（包括系统扩容、压测、监控、后端优化、安全、应急演练），并由平台架构师牵头负责对这些任务的实施和整体把控。

◆ 对整个购物流程都做了非常细致的应急预案，通过降级个性化服务来保证用户能够快速访问。例如，在支付环节要依赖京东台账和订单中心，如果服务出问题，通过降级保障用户能正常支付。即使通信光缆中断等极端情况发生，也可以通过平台的跨机房多活、流量入口的转移、应用系统的跨机房切换，以及有效的限流和降级等机制，来保证用户的购物体验。

◆ 建立了以运维、云平台、网络和系统性能为基础的多维度监控体系。从底层的操作系统的各项指标，到各个层面的业务系统，再到网络都需要监控，这些监控一起形成了一个比较立体的监控系统，从底层到最上层都有监控。

2017年的京东618大促正值印度尼西亚斋月，活动期间日均订单环比上涨137.7%，同比上涨1031.76%，6月10日达到了订单新高峰，创印尼电商平台单日订单量历史新高。京东海外研发的技术体系演进总结为初期的平台搭建，中期的驱动业务发展，未来的引领企业成长。通过618大促备战和应战，京东海外平台研发部的技术体系优化得更加稳定，能从容应对突发的压力，同时积累的大量运维经验和应急预案也让整个技术部门受益匪浅。

6.4.2 营销产品技术的变革

营销创新为京东业务增长提供了原动力，在保障业务的增长的同时引领业务增长，需要的是不断进行的产品创新和技术变革。本节将揭开京东搜客流量引擎、JShop装修及营销平台、会员营销系统这些主要营销产品的神秘面纱，让您看到快速增长和复杂多变业务环境下的营销产品技术演进之路。

1. 搜客——汇聚八方来客

搜客平台承担了京东站外搜索引擎引流的重任。为京东引入越来越多的优质流量，是搜客平台的使命和责任。

◆ **系统代替人工——流量引擎1.0。**大促期间，如何安全高效地引入优质流量，是京东网络营销的头号任务，基于这个目的，对接搜索引擎的搜客平台应运而生。

搜客平台诞生前，要想从搜索引擎引流，需要人工在各搜索引擎后台手动操作关键词的购买，并且人工定义流量如何分发到站内活动页，人工操作工作量巨大，每次大促都是一场肉搏战，因为人工操作的局限性，引流效果往往不尽人意。

2016年5月，搜客平台一期正式推出，系统对接了国内各大主流搜索引擎，打通各搜索引擎API，可以跨账号管理千万级关键词的竞价，将物料自动上传，并且提供竞价效果跟踪，支持快速优化，彻底解决了人工操作的效率局限。

搜客1.0使我们从人工的重复操作中解放出来，将更多的精力放在了思考如何引入优质流量上。



◆ **操作无须等待——流量引擎2.0。**2016年下半年大促，市场竞争异常激烈，运营不断调整关键词出价，以维持流量引入，搜客开始出现性能瓶颈。通过紧急增加部署，扩充上传线程，把上传性能提升了3倍，总算有惊无险。

随着业务的不断发展，搜客平台处理和存储的数据量越来越大，系统的瓶颈也越来越明显。每次批量修改关键词的出价、落地页，任务排队导致需要等待很长时间才能生效，并且经常因为数据量太大而无法得到效果数据。

考虑到大促期间系统需要保持的高可用和快速响应，采取了小步快跑、逐步演进的做法。搜客2.0重点提高大数据的读写能力，鉴于每次操作都可以涉及百万级的数据变更，采用ES集群替代Solr，提升物料实时修改能力，并且引入了Druid、Spark、Hive等大数据处理技术，解决实时报表查询能力。

同时，业务快速扩张，导致平台中存在越来越多的无效物料，无效物料会导致用户在点击搜索引擎广告时，进入的页面错误或活动已过期，因此，搜客2.0实现了对自身进行体检，可根据场景自动识别违规物料，识别过期活动，剔除无效物料。

搜客2.0解决了亿级数据批量处理的时效性，让每一次操作无需等待，同时让亿级物料自动体检，提升了广告的用户体验。

◆ **智能引流未来——流量引擎3.0。**经过一年的演进，搜客平台实现了从无到有、从有到快的发展，随着业务不断扩张，智能引流的3.0时代已迫在眉睫。

搜客3.0将通过系统自动获取关键词，自动匹配着陆页，自动调整关键词出价，并且接入更多更丰富的广告样式，让搜索引擎营销真正进入智能时代。

2. 百变装修，个性化页面装修的缔造者

◆ **多样化需求下的JShop开放平台生态。**JShop是京东最大的页面装修、发布平台，作为具有京东页面装修标准的自有软件著作权产品，从2012年诞生至今，已发展为覆盖PC端和移动端店铺、活动、商详等页面创作的全能型装修平台，这不仅有技术架构的不断演进，也有产品平台生态的逐步完善。

JShop平台现在已支持京东80多个二级部门，支持京东商城、京东金融、京东到家等前台页面搭建业务，其覆盖的用户群体不仅有内部的采购、运营，还有数以万计的商家用户。在这样的业务背景下，衍生了多样化的、个性化的甚至是临时性的装修需求，摆在这个产品发展面前的第一个问题是如何通过几十人团队的研发资源来支撑如此巨大体量的页面装修业务。

2013年，JShop发布了京东模板市场标准化的SDK和API，提出开放平台的思路，统一了开发语言，统一了接口标准，进行了多次针对第三方设计师的开发培训，形成了京东自有的装修市场模板规范，使得第三方设计师可以高效地开发出高质量的模板，同时也有效地解决了安全问题。通过建立模板市场，除自己研发的装修模板、组件以外，JShop装修平台还可提供万余套优秀第三方设计师页面模板和丰富的装修组件，并提供各类定制化装修服务，进一步增强了京东各类营销页面的视觉呈现能力和页面营销玩法，解决了海量多样化页面装修需求下的供需平衡问题，也降低了临时性需求的研发支持成本。当然，如今的模板市场已成为京东ISV（Independent Software Vendors）最活跃的平台之一，并为京东完

成了交易创收。

解决了装修业务拓展和装修效果提升问题，JShop面临的另一个关键问题是如何帮助装修用户进行营销页面数据的跟踪和装修分析，并进一步形成智能营销策略指导。为了形成装修到数据的产品闭环，2016年，JShop开始完善平台的数据可视化服务产品体系，通过用户画像、实时数据计算、大数据建模与推荐算法、热力图技术，形成了涵盖数据报表、装修分析、数据诊断、实时监控等可视化产品体系，同时完成了千人千面的智慧店铺等大数据应用创新，通过针对不同特征的人群展示不同的页面不同的素材，有效提升了营销页面的转化率。

如图6-31所示，JShop已形成以京东活动、店铺等装修系统为核心，模板市场和数据中心进行装修赋能的装修平台产品生态，为历年京东618大促和日常各类营销页面装修、发布、浏览保驾护航！

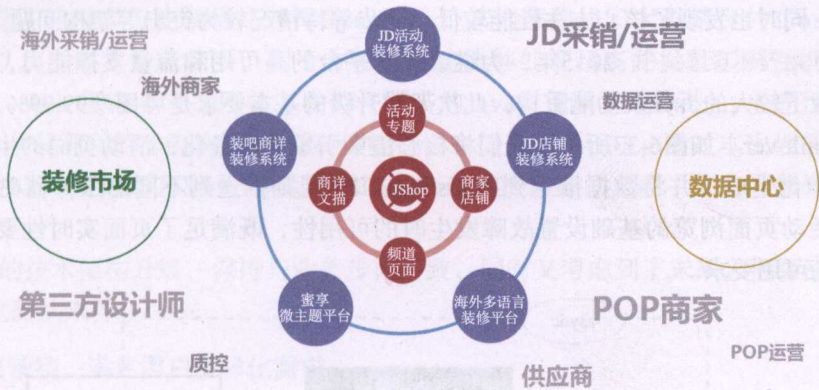


图6-31 JShop装修平台产品生态

JShop亿级流量技术架构演进之路（图6-32）

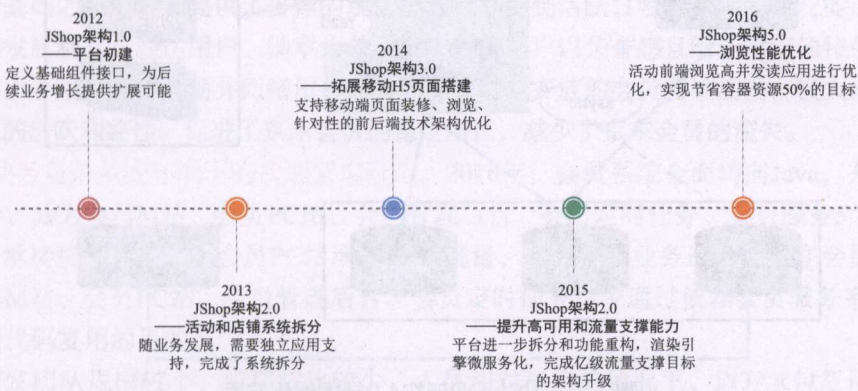


图6-32 JShop技术架构演进之路

◆ JShop技术架构1.0。2012年，JShop平台开始初期建设，目标明确，为京东提供一个搭建页面的平台，满足采销运营搭建活动页面和自营店铺页面的需求。平台建设初期，结合业务现状，系统做到简单、易于开发维护、满足基本业务需求即可，因此定义了基础组件接口，为后续业务爆发式增长提供扩展可能；采用Memcached缓存热点数据、页面组件



片段数据；自定义缓存Filter，在Filter中直接读取 Memcached热点数据进行页面组装。

◆ **JShop技术架构2.0。**2013年，随着公司的不断发展，外部商家开始接入平台，形成了两条较为清晰的业务线：自营和POP商家。为了更好支持POP商家的业务需求，系统也水平拆分为JShop活动平台和JShop商家平台两大独立平台。由于对更多业务的支持，我们将应用按功能职责服务化，拆分为多个子应用，例如装修平台、浏览平台、服务中心、数据中心、管理端和配置中心，将基础设施操作部署为公共服务组，公共RPC服务调用抽象，用于各子应用依赖。

◆ **JShop技术架构3.0。**2014年，移动互联网时代到来，JShop 3.0要求能支持移动H5页面搭建。经过近2个月的紧张研发，我们成功打造了京东第一款移动端活动装修平台，支持京东App和M站上的活动页装修及浏览。在这之前，由于团队都未涉足移动端相关产品的研发，一开始活动端页面在各方面的体验都饱受诟病，页面浏览性能、流量消耗都引来了较大的争议，同时也发现了核心方法性能较低、同步等待情况较为突出等架构问题。

◆ **JShop技术架构4.0。**2015年，为继续提升平台的高可用和流量支撑能力，我们再一次对平台做了较大的拆分及功能重构，此次架构升级的基本要求是可用率99.99%、10亿PV/天、秒级Failover。如图6-33所示，我们将核心渲染引擎微服务化，活动页面的生成通过定时任务异步化生成，并将数据推送到Redis和按URL规则推送到不同的服务器磁盘用于备份，满足活动页面浏览的基础设置故障发生时的可用性，既满足了页面实时性要求，又满足了系统高可用要求。

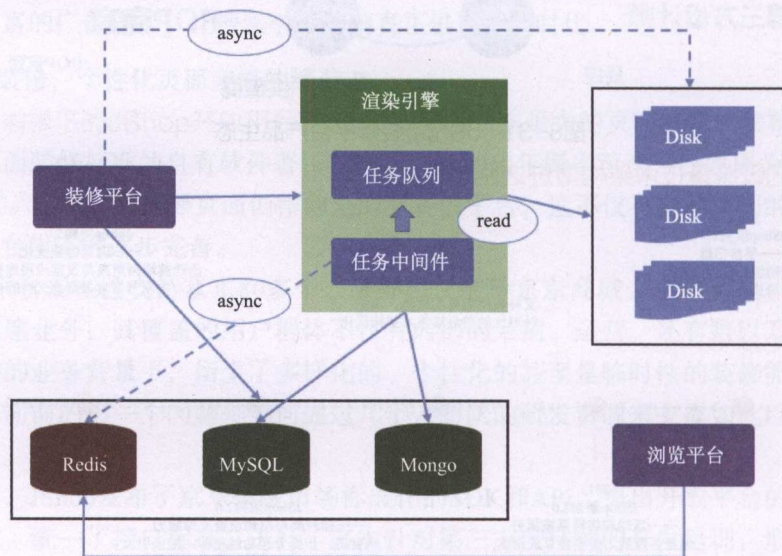


图6-33 JSHOP架构4.0技术优化方案

◆ **JShop技术架构5.0。**JShop 4.0经过一年的无故障稳定运行，核心基础组件稳定性和可用性得到了有效验证，达到了设计目标，线上服务器已全面虚拟化，但容器资源的申请依然不是件容易的事情。同时，新的业务不断出现，而服务化拆分不得不使用更多的资源。在这种情况下，部门内自给自足成了一种优先的解决方案，将老应用缩容以满足新应

用的资源要求。那么，老应用怎样才能在资源减少的情况下，保持不变的服务能力？答案就是提升性能。

如图6-34所示，选择了活动前端浏览高并发读应用进行优化，经过压力测试，单个4核/8GB规格容器，QPS只能达到750

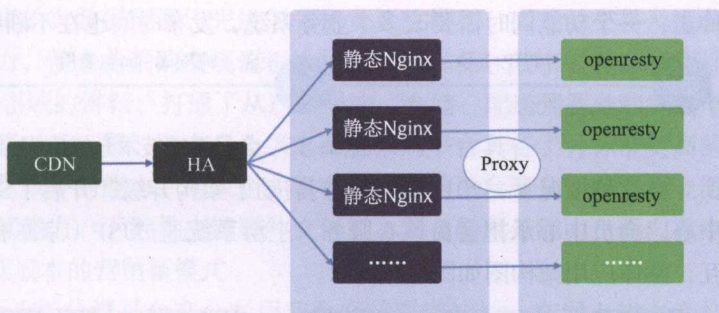


图6-34 JShop架构5.0技术优化方案

次/s左右，考虑到活动浏览应用属于I/O密集型高并发读应用，业务逻辑较为单一，应用也比较轻量，经过多方比较最终选择Openresty，其内部集成了大量精良的 Lua库、第三方模块，以及大多数的依赖项，用于方便地搭建能够处理超高并发且扩展性极高的动态 Web 应用、Web 服务和动态网关。我们将原来的Java应用通过Openresty来进行重构，充分利用Nginx本身利的非阻塞I/O模型高性能特性来提升应用QPS，通过本次性能优化，相同配置下，QPS达到3000次/s，达到原来Tomcat应用的4倍，实现了节省容器资源50%的设计目标。

JShop的技术架构升级，保持与业务步调一致，同时又考虑到了未来。亿级流量技术架构的演进之路从未停止……

3.会员营销，满足用户多样化需求

◆ **业务发展——多维度活跃用户。**京东会员频道通过为不同身份的会员提供差异化权益，包括固定权益、异业福利，强化会员对于京东身份的认知，促进用户在京东的活跃度。在权益的基础之上我们还提供了多样的玩法，如京豆签到活跃日常用户、礼包智能选时选券选人精准发放利益点触达用户、勋章为各品类高价值用户以荣誉感且给予对应的特色权益、任务中心引导用户购物互动并回赠用户利益点等。百花齐放的会员营销活动，极大地提高了京东会员的活跃和黏性，促进了京东会员的稳定增长，减少了京东会员的流失。

◆ **业务复杂多变环境下的快速架构升级。**2016年，会员系统全面转向Java，并进行了应用拆分，分为会员M站、会员PC站、会员管理后台、会员定时任务、会员服务5个系统。会员M站承接移动端流量，会员PC站承接PC端流量，会员核心业务逻辑建设在会员服务系统，会员M站、会员PC站、会员管理后台、会员定时任务系统通过依赖会员服务系统的Jar包来达到代码复用的目的。

在研发团队规模较小、应用数比较少、人员相对稳定的情况下，以JAR包复用代码的方式是能起到比较好的效果的，JAR包复用避免远程RPC调用的开销，应用数较少的情况下测试发布成本也不高。这种方式对人员要求较高，需要大家有判定系统边界的能力并遵守规则，但在业务逐渐复杂，团队人员规模扩大，系统演进因拆分而变多的情况下，很难保证JAR中的代码是一些公共服务，结果就是公共JAR中的代码会越来越臃肿，越来越庞大，包含各种业务，系统间耦合性越来越高，同时系统数量的增加也对研发效率造成了巨大的



冲击，一个功能同时需要改多个业务系统，发布时间也在不断膨胀，如图6-35所示。

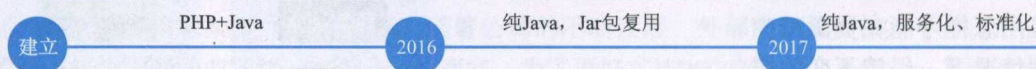


图6-35 会员系统技术升级历程

在这种情况下，2017年6月我们启动了架构升级，开启了SOA服务化的时代，建立会员中心，会员中心承担会员核心服务，上游系统通过JSF（京东自研RPC框架）和会员中心交互，整体应用架构图如图6-36所示。

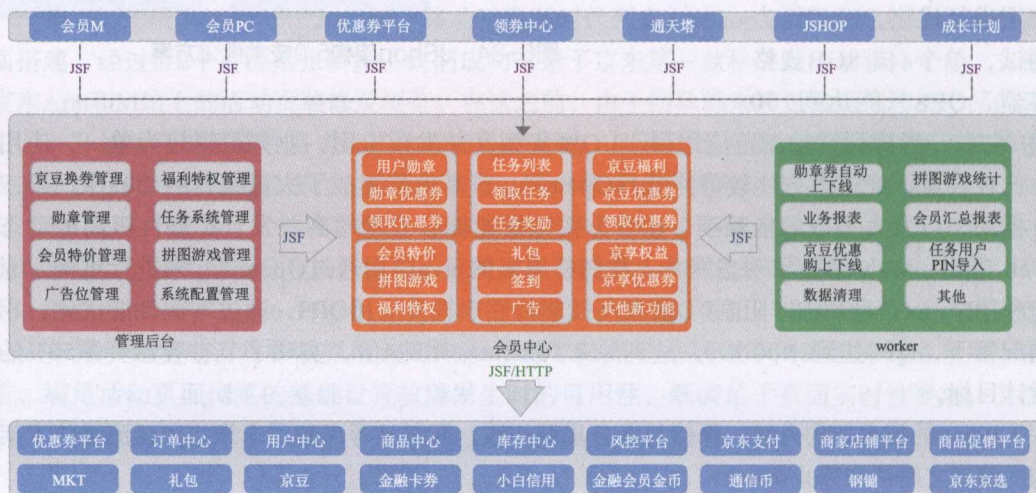


图6-36 会员中心整体应用架构图

我们将开放更多有趣的玩法和更丰厚的利益点回馈产品。在此基础上将完成SOA服务化的改造，提高系统的稳定性和可靠性，为业务的快速发展打下坚实的基础。具体从以下三方面进行升级：

- ◆ **精细化运维。**建立完善的系统监控、业务监控、完善的性能分析工具，快速发现问题、解决问题。
- ◆ **最终一致性。**避免使用分布式事务，通过最终一致性保证业务正确性，提高性能。
- ◆ **异步化。**对优惠券等促销工具类业务，大促期间往往会迎来爆发性流量增长，领取优惠券业务处理复杂，交互系统多，通过引入MQ削峰填谷，应对流量增长。

6.4.3 京东B2B征程

京东B2B业务深耕B2B电商行业，通过供应链服务的方式帮助中小企业实现互联网化的转型。

1. 系统介绍

◆ **企业级电商化采购平台。**企业的生存和发展离不开一个高效阳光的采购团队，传统意义上的企业采购，包括采购组织、采购过程管理、采购审计、采购结果评估等多项内容。

如何使整个企业采购过程能快速流转、商品全国可选、价格可比、全程阳光透明？如

何借助电子商务平台海量健康的大数据资源、先进的分析及算法模型、高效的交易及运营能力、优良的供应链服务能力，使企业采购实现流程重构、成本降低、效率提升、管理精准？京东致力于自营式电商领域的耕耘，打通了从产品采购、仓储、配送到服务的完整电商价值链，通过建设“智、慧、云、翼”四大企业级电商化采购平台具备了为标准更高的工业企业市场提供采购电商化服务的能力。四大企业级电商化采购平台聚合京东自身上游供应链服务能力、大数据计算能力、下游供应链履约交付能力及金融服务能力，面向重点行业市场提供高效、阳光、低成本的营销新模式。

◆ **企业级分销平台。**企业级分销平台充分利用互联网的渠道特性，在网上建立产品分销体系，通过网络把商品分销到全国各地。通过分销系统，企业可以节约很多成本，并且利用有限资源，跨过时间、地域限制获得更多利益；拥有充足的空间和市场进行品牌宣传、产品推广；网上分销可以和消费者直接接触，减少中间环节；掌控价格市场，避免价格战及不良竞争模式；开发各种渠道及代理，扩大分销市场。

2. 京东B2B系统的演进

平台的发展是随着京东电子商务平台业务同步发展的，由原来的冒烟式构架、集中式存储与计算逐步演变为基于微服务、模块服务的分布式计算架构，演化出商品、订单、数据、发票、用户、促销等基础模块平台，在多个基础模块的基础上封装业务功能通用服务并提供Open-API，前段页面进行页面组件化升级，对个性化需求进行可配置，在Open-API基础上构建基于插件的SPI开发生态圈，整体演进思路如图6-37所示。

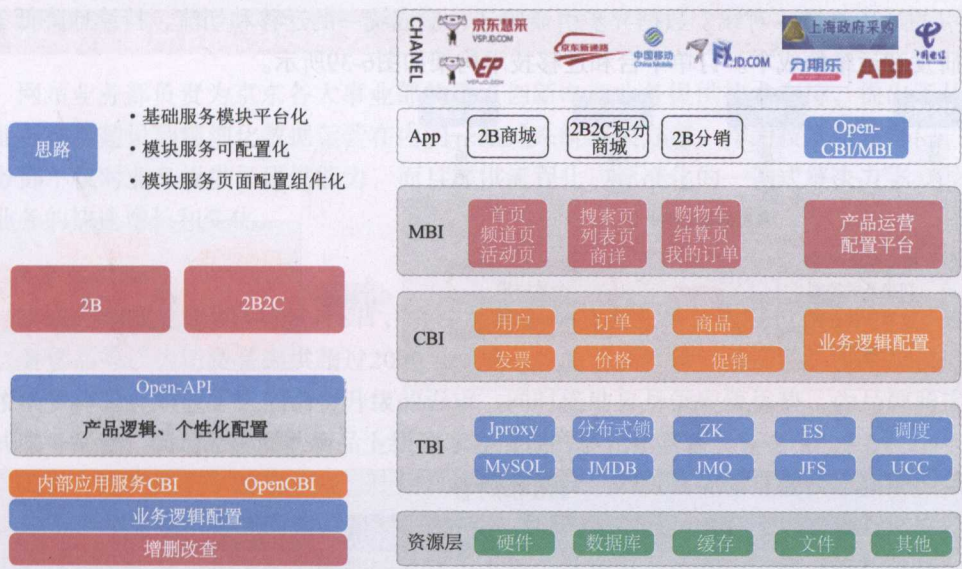


图6-37 B2B系统整体演进思路

3. 技术支撑

◆ **京东B2B技术基础。**B2B企业平台构建在京东云和京东中间技术之上，底层资源层具备虚拟化，动态水平扩展，中间件技术提供了分布式消息中间件JMQ、分布式文件存储



JFS、分布式缓存集群JMDB、分库中间件JPROXY||&GUN、Storm、HBase、RPC框架JSF等基础技术，将通用技术服务工具化，如分布式锁、调度、bin-log同步、分布式内存一致性、统一配置中心等。

◆ **京东B2B基础模块平台化建设。**2014年年初，京东B2B系统封装了京东基本数据服务的接口组件成功上线，当时只有几十个客户。2014年年底，服务企业数已经达到了几十万用户数，不仅给系统带来了不小压力，同时因企业客户需求的多样性和复杂性，使得系统在性能和扩展性上同时遇到了巨大瓶颈，快速响应系统业务需求对系统的分层和划分的思考不足体现了出来。从2015年开始，逐步对系统进行升级优化，首先是将基础模块进行平台化建设，将企业购商城、企业分销商城、积分商城的基础模块进行平台化建设。

基础模块平台项目分三期实施，三期项目都带有明显的业务目标，以业务目标为驱动的架构演进方式也成为B2B后续很多项目实施的参考。整体平台化时间轴如图6-38所示。



图6-38 B2B基础模块进程

订单平台化项目主要有两个亮点：第一，通过抽取电商公共元素订单模块，沉淀了共享服务，降低了重复、创新和试错成本，奠定了京东B2B系统架构升级的基础；第二，形成了一套分库支持底层数据存储水平扩展的技术方案。基于bin-log统一异步建立缓存和搜索索引的通用系统，对线上数据异步引流验证，形成统一的迁移和切流，持续地降低了学习、研发和运维的成本。订单平台和迁移技术构架如图6-39所示。

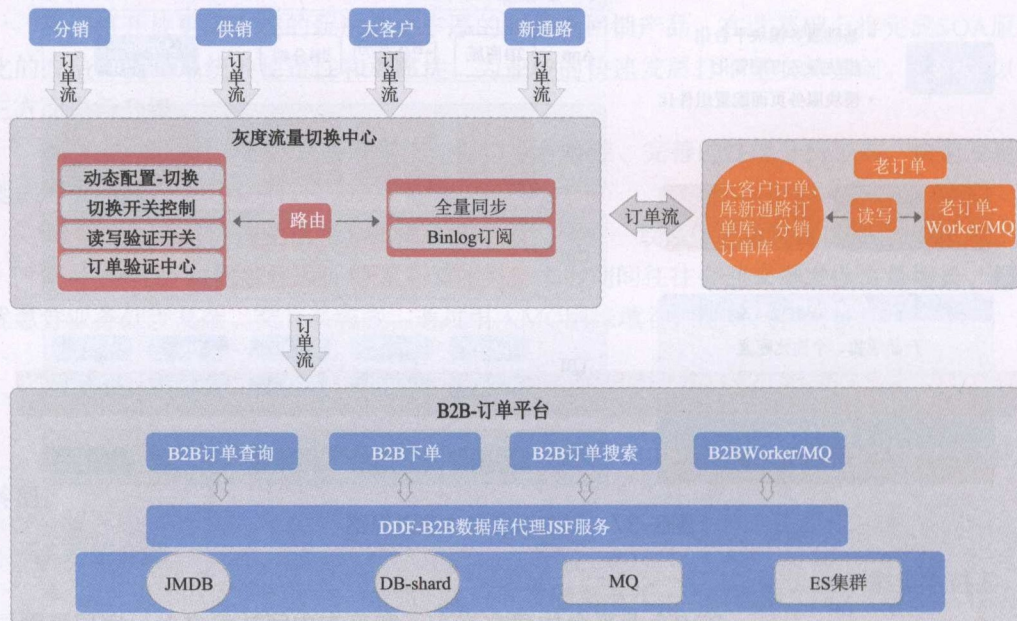


图6-39 B2B订单基础模块构架设计图

通用订单服务平台设计是以基础组件服务平台化的思维来构架的，内部模块具备高可用、高并发、易扩展的特点，平台内部分成四大模块，订单中间件模块服务完成订单相关的读服务，下单服务模块提供可配置的话的订单XML对外灵活的参数定义，将主流程和非主流程分离，将读写服务分离，将系统间的通信异步化，将调度中心化，将接口对接参数可配置化，做到灵活组件可插拔。

企业采购业务、系统的并发和订单的量可以达到亿级别的存储需求，需要对系统按订单模块来垂直拆分，满足业务增长的需要，对订单按照企业号来划分库，其他ID映射企业号来查询，对于订单量大的企业做特殊路由分库，将历史数据进行归档，跨库查询使用ES，通过任务冲正、消息重试、消息二阶段提交等来保障数据一致性，将运用业务逻辑和数据访问分开，隔离了对底层数据库资源的使用，部署上更具扩展性。

分库之后，表的唯一主键就不能使用MySQL的auto increment了，需要生成全局唯一的主键。为此，我们自主开发了基于JVM内存ID队列、H2嵌入式数据库ID缓存、MySQL表ID状态存储的全局ID生成器。

◆ **京东B2B组件化建设。**在系统组件化方面，首先是开票系统的组件化，思路是获取业务系统的头尾和菜单嵌入到开票的页面中，形成通用的组件，更进一步基于基础模块平台化和功能模块服务构建完整的组件，包含模板、多语言、多主题，在组件的基础上通过各个组件的编排形成通用的业务运用，如2B2C商城、积分兑换商城，在运用基础做产品化配置形成场景化的产品，如微信、生活杉德、惠采、翼采等。

6.4.4 重点创新电商业务

网站业务部负责为京东各事业部的重点创新电商业务提供技术支持，提出了从差异化业务系统建设到精细化数据运营在内的一站式系统解决方案。作为技术研发部门，网站业务部不仅对业务提供了研发能力，而且提供流程化、标准化的一站式解决方案，及时响应业务的快速增长和变化。

1. 京东闪购

京东闪购成立于2014年1月8日，专注品牌限时特卖业务，内容包括女装、男装、居家、奢侈品等，为消费者提供超过2000个优质品牌的正品特价专场活动，带来高质低价的愉悦消费体验，顺应了人们消费升级的诉求。同时借助自身供应链优势，为品牌商提供一站式服务平台，满足品牌商从新品上线到库存清理的多元化需求。

秉承“用户至上”的价值观，让用户在闪购平台上购买到优质且性价比高的商品，于2016年下半年启动了智能闪购。初期主要是围绕闪购活动分析和基于用户的智能推荐，其中闪购活动分析是对每天参加闪购的，成千上万个活动以及活动中的商品进行分析。根据活动的曝光情况，结合每种曝光行为的权重，采用机器学习算法，为活动及活动中的商品进行打分，得分越高说明越受欢迎。图6-40是闪购活动分析抽象图。

闪购智能推荐系统负责把最想看到的内容推荐给用户，前期是利用闪购活动分析系统给出的评分，把优质的活动推荐到最显眼的位置。之后系统会自动收集推荐效果相关的数据，用于推荐模型，来不断地优化推荐模型。技术上使用Hadoop+Spark作为主框架，评价



分析采用NLP的相关技术，同时对多种模型效果进行对比优化，针对不同用户选出适合的模型。图6-41是该闪购智能推荐抽象图。

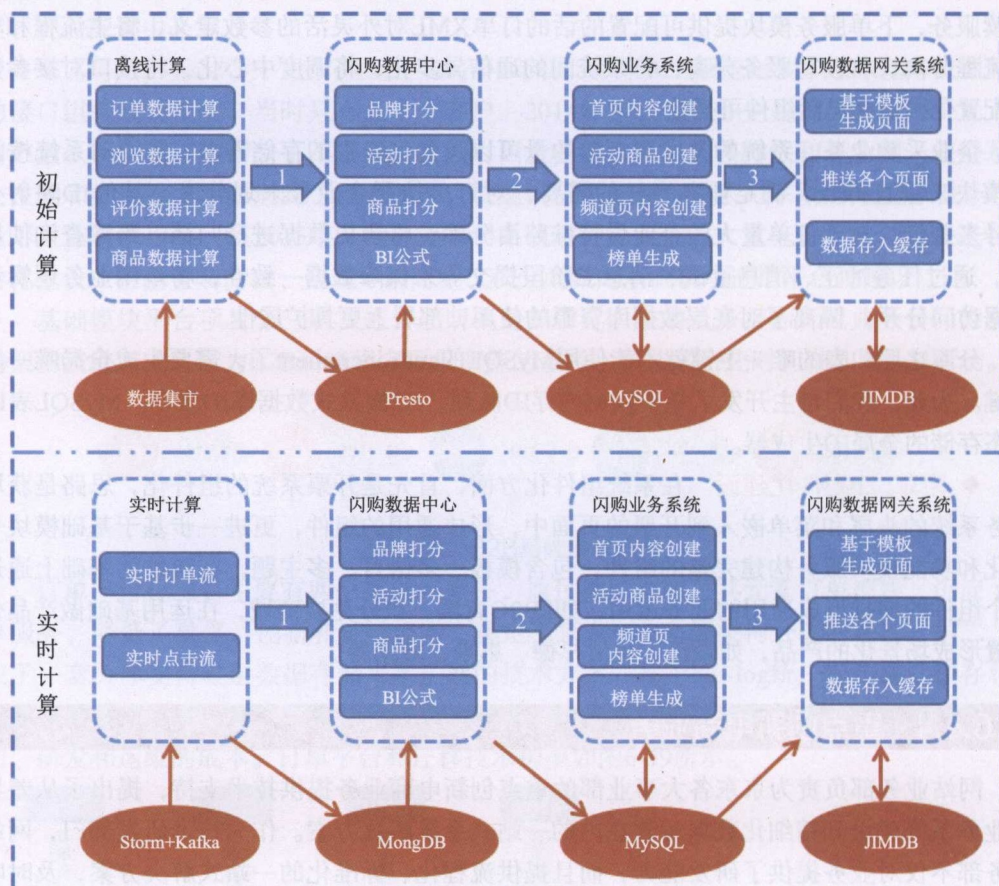


图6-40 闪购活动分析抽象图

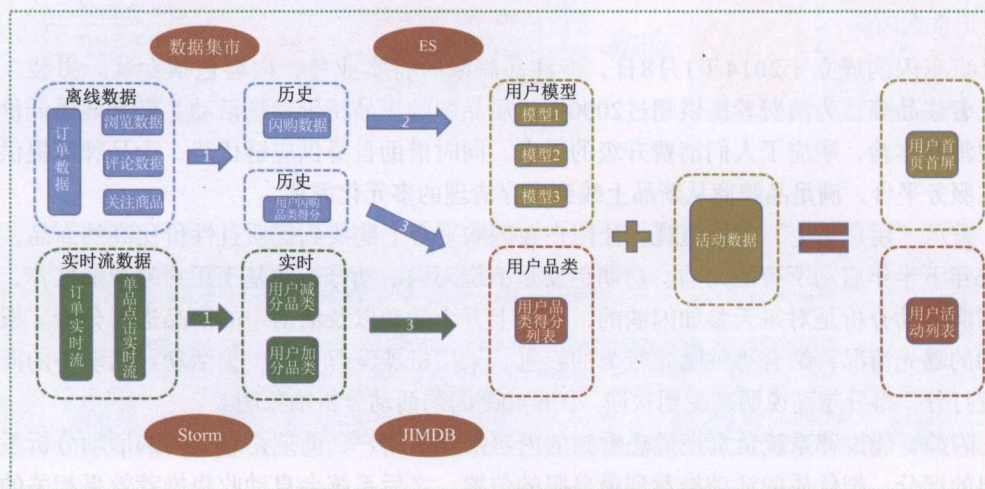


图6-41 闪购智能推荐系统抽象图

2. 京东全球购

在电商全球化的浪潮之下，作为国内领先的电商企业，京东全球购承载了京东人走向世界的梦想，在这个全球化背景的驱动下于2015年4月正式上线，揭开了京东全球化的序幕。在具备通用的电商模式、购物流程外，京东全球购还有着与国内电商不一样的关税、物流配送、售后等流程，同时在网络上还面临着海外部署的挑战。

全球购平台建设之路。首先基于现有的基础平台规划了整个全球购平台，打通整个购物流程并覆盖全渠道，同时完成海关及服务商系统的对接。当基础平台上线之后，又扩展了两种电商模式，实现了B2B、B2C、C2C多种模式全覆盖，同时实现了包括保税备货、保税集货、直邮、个人快件等多种国际物流模式解决方案，在支持业务扩充的同时进行系统提升和改进。整体系统架构如图6-42所示。

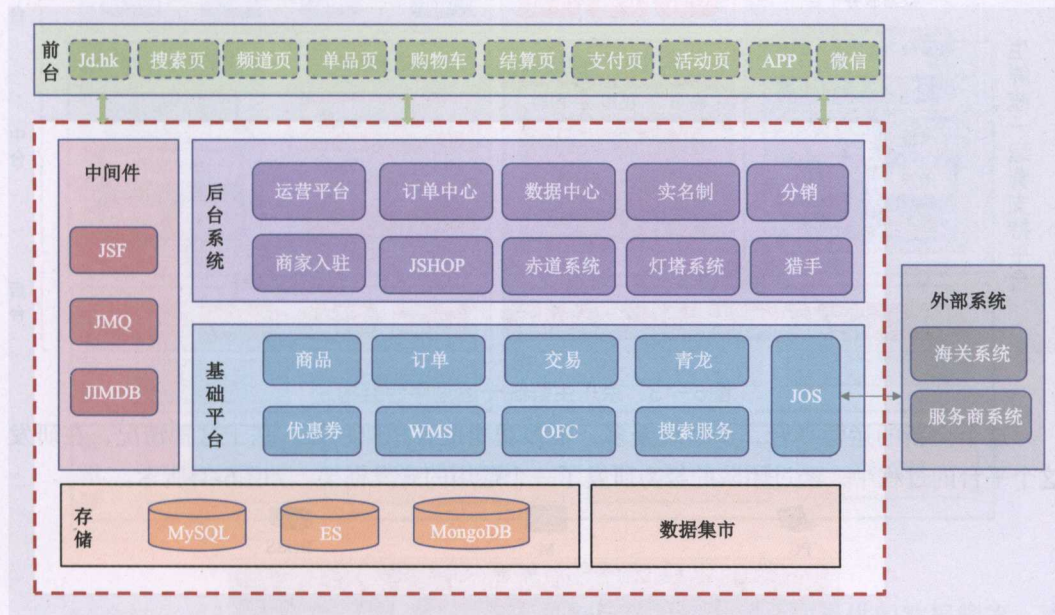


图6-42 京东全球购整体架构

可用性保障。为保障全球购这种级别的门户网站的高可用性，针对系统架构从以下几个方面进行了改造和提升。

- ◆ Nginx、JIMDB、JVM多级缓存，以及数据托底，确保门户网站的高可用。
- ◆ 双机房部署，防单点，确保每个机房能独立承担百分百流量。
- ◆ 限流，以确保流量冲击时，系统高可用。
- ◆ 依赖服务自动降级，避免非必要功能影响系统可用性。
- ◆ 重新规划PC、M、App，不同平台应用彻底解耦，去除相互影响。

系统性能保障。为了提升系统性能，在对网络部署、应用规划、架构升级等方面进行全方位改进的同时，还做了大量看似不起眼实则对性能影响很大的微改造，如数据缓存、内容压缩、页面异步并行加载等。



3. 京东生鲜运营平台

经过一年的发展，京东生鲜已有一定的业务基础，也积累了一部分忠实粉丝。但在生鲜电商发展如火如荼的这两年，面对残酷的竞争，如何让京东生鲜运营达成业绩，提升GMV将是一个挑战，为此专门为京东生鲜打造统一运营平台。

初期统一运营平台主要聚焦在运营数据，提供多维度的数据统计分析模型，全方位多层次数据报表，同时为京东生鲜用户构建用户画像，为运营人员提供数据参考，后续将再朝着智能化运营的方向扩充。同时，为了提升京东生鲜的业界影响力，带来新的用户，引入了“东家菜”模块，给用户提供各种菜谱和做菜视频，供用户学习。统一运营平台业务架构如图6-43所示。



图6-43 京东生鲜统一运营平台架构

这个全新的运营平台，子系统繁多，很多功能也有相似之处。基于这种情况，在研发这个平台的过程中，经过团队的努力研发了一个通用的开发框架，如图6-44所示。

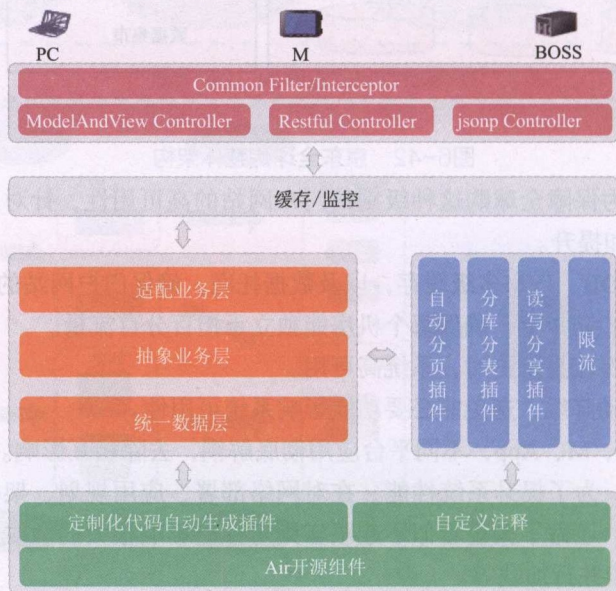


图6-44 统一开发框架

对于开发人员来说，搭建开发框架是基本要求，但此框架的不同之处是除了基本的业务抽象、数据抽象，引入缓存外，还具备Web开发常用通用的数据读写分离、分库分表、限流相关解决方案，以插件的方式存在于框架中，同时引入了内部开源组件Air。Air包含了大量的功能组件，高并发分布式锁、分布式任务调度及实时监控等。开发人员在使用本框架时，只需要关注业务本身的实现逻辑，并在不同的场景下选用适当的组件即可，极大地提升了开发效率和系统的可维护性。

4.智慧数据运营

无论是京东闪购、京东全球购、京东生鲜还是其他业务，成功的运营都离不开数据的支撑和工具的协助。于是，在2015年网站业务部开始着手打造基于运营体系的数据和工具平台来支持新业务的快速发展，图6-45展示了这些数据工具。

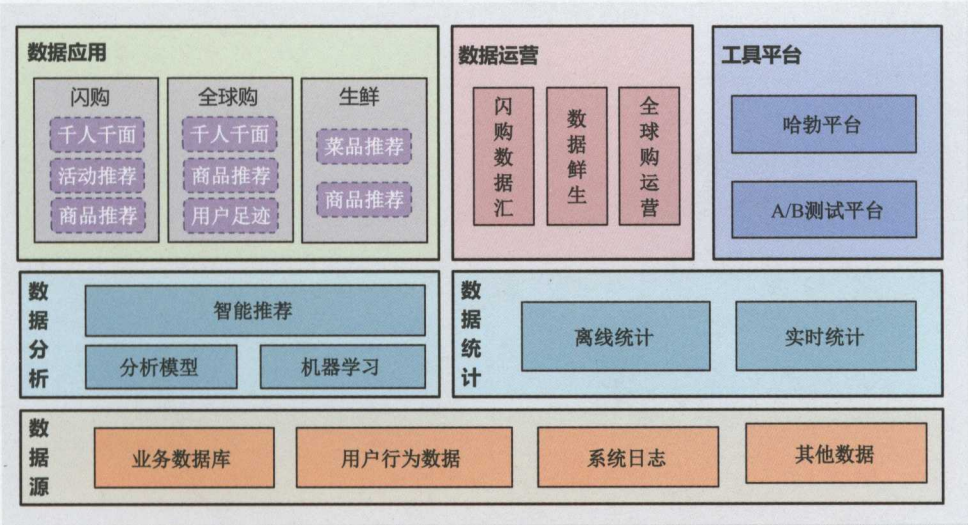


图6-45 智慧数据运营工具

◆ **个性化数据统计——解决燃眉之急。**初期运营是通过在公司提供的数据管家、流量可视化等平台获取统计数据，由于这些平台都是为全公司服务，只提供通用的标准化数据，所以从这里获取的数据要么统计维度比较粗犷，要么数据指标缺失。若是对各自经营的类目、品牌、SKU等从多维度全方位进行数据分析，只能通过BI进行提取数据，不仅耗时费力，效率还不高。基于这种情况，我们有针对性地为事业部运营提供个性化的数据统计、报表服务，先后建立了闪购数据汇、数据鲜生、全球购运营数据平台，以服务这几大事业部的运营。

◆ **运营工具平台——为运营决策铺路。**为运营解决数据统计分析之后，还提供了数据深度分析的哈勃平台和版本改进效果跟踪分析的A/B测试平台。哈勃平台提供多个主题（例如生鲜订单主题、生鲜SKU主题、全球购用户主题等）的多个维度，以及丰富的指标数据，数据分析人员除了能进行基础的自定义报表分析外，还能针对维度进行数据深度钻取和主题多维度联动分析。针对运营的活动、商品推广等，运用A/B测试平台进行版本投放，收集不同版本的访问、下单、订单转换率等数据，计算置信区间，分析各个版本的投放效

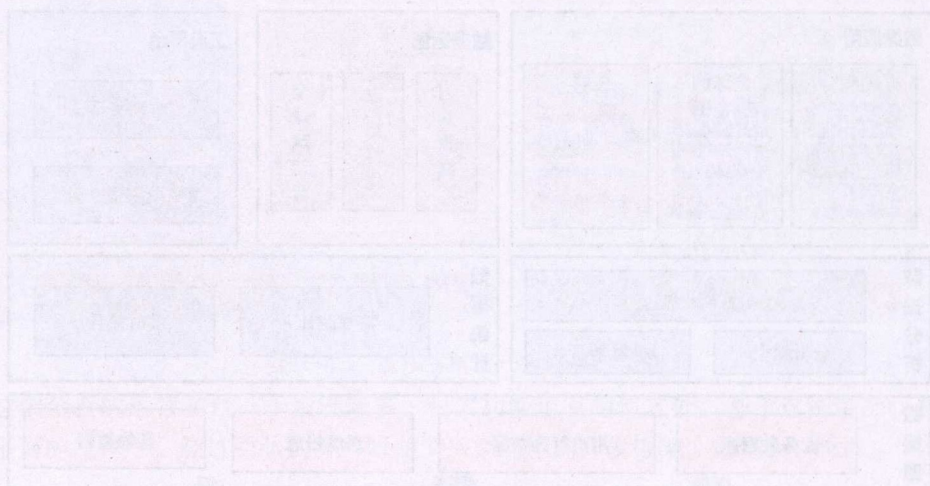


果，为运营决策做参考。

◆ **智慧数据运营——自动化精准运营。**数据运营是基于历史经验数据的多维分析，运营人员只能根据该数据制定宏观决策。当业务发展到一定阶段，只依靠宏观决策并不能很好地进行业务拓展，此时自动化精准运营悄然而生。2016年年底，已经着手构建全方位用户画像，运用机器学习，分析用户偏好、预测用户行为等，提前干预用户购物行为。目前，自动化精准运营已经在闪购、全球购和生鲜业务里运用，后续也将是主要的发展方向。



飘雪一郎：营销创新研发部架构师，京东活动平台负责人。
Jerry：B2B业务部架构师，负责部门系统构架和备战工作。

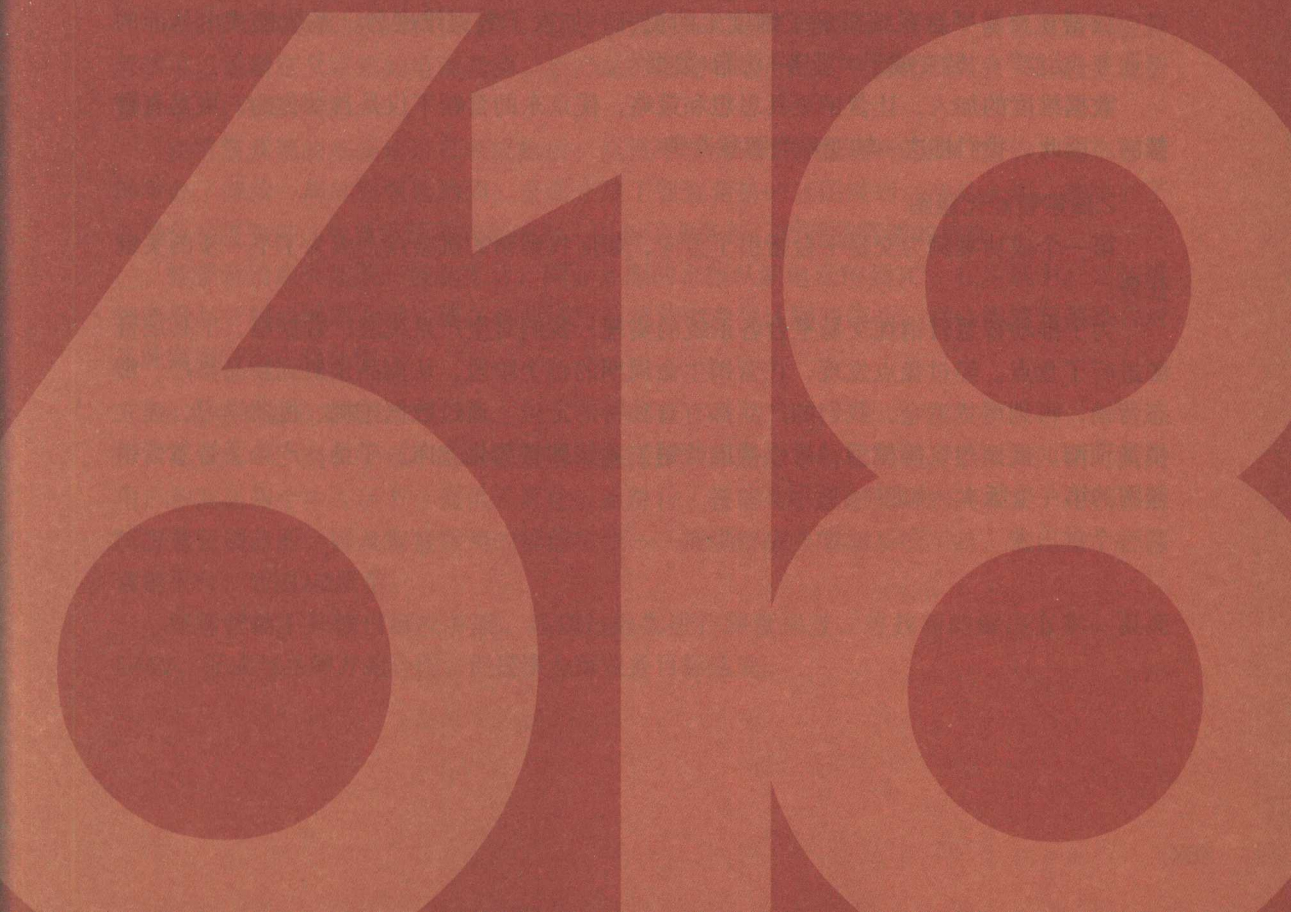


第7章

智慧营销

知你所想，予你所求

前事不忘，后事之师。京东在多年的经营之中，已经积累了庞大的电商交易数据。利用大数据技术结合人工智能、机器学习，京东正在进行全面的智能化改造，打造新型智能电商生态。





7.1 智慧营销

7.1.1 智慧营销平台的前世今生

1. 从困局窥见新生

京东交易平台（以下简称“交易平台”）隶属于京东商城研发部，负责京东的生产流程、交易流程、用户管理与营销工具的基础功能研发，交易平台在京东各技术研发部门中，以技术储备深厚、产品研发高效、系统性能可靠著称。在历年的618中，交易平台承载交易洪峰，提供了最可靠的技术支撑。

2016年初，交易平台也在思考自身的发展困境：沉淀多年的基础交易与营销工具，如何突破传统，焕发新的活力？

数据成为突破口，第一款产品——抢客神器的出现，让交易平台看到了数据带来的新生。抢客神器通过对交易数据的机器学习，找到高潜用户，并结合促销工具对高潜用户进行定向营销。与传统促销模式相比，这一创新效率与效果得到大幅提升。这一探索也加速了交易平台对系统数据化的思考。

交易平台有着多年的交易生产数据沉淀，庞大的数据像一座金矿，大数据技术的发展，使发掘这座金矿成为了可能。通过数据挖掘和机器学习，把生产系统的庞大数据与营销工具结合，重构营销流程，成为了交易平台新的方向。

数据让交易平台系统得到了维度上的提升，加入了数据维度后，系统模式由从前的“业务+功能”，转变成了“业务+功能+数据”。

数据维度的加入，让营销更具思想和策略，使京东的营销不仅是自动化的，更是有智慧的。因此，我们将这一转变称为智慧营销。

2. 探索智能化蓝图

第一个成功案例为交易平台指引了方向，如何找到新的机会点，成为了下一步的关键任务。

为了推动智慧营销在交易平台各系统的实现，我们对生产系统层、数据层、工具层资源进行了盘点。经过盘点发现，在营销生命周期的每个阶段，从商品预研到新品推广、常态营销，再到尾货清仓，我们的产品都有着独特的定位。通过数据挖掘、机器学习，建立销量预测、促销建议等模型，可以帮助营销工具找到智慧化方向。于是，产生了智慧营销蓝图的第一个版本，如图7-1所示。

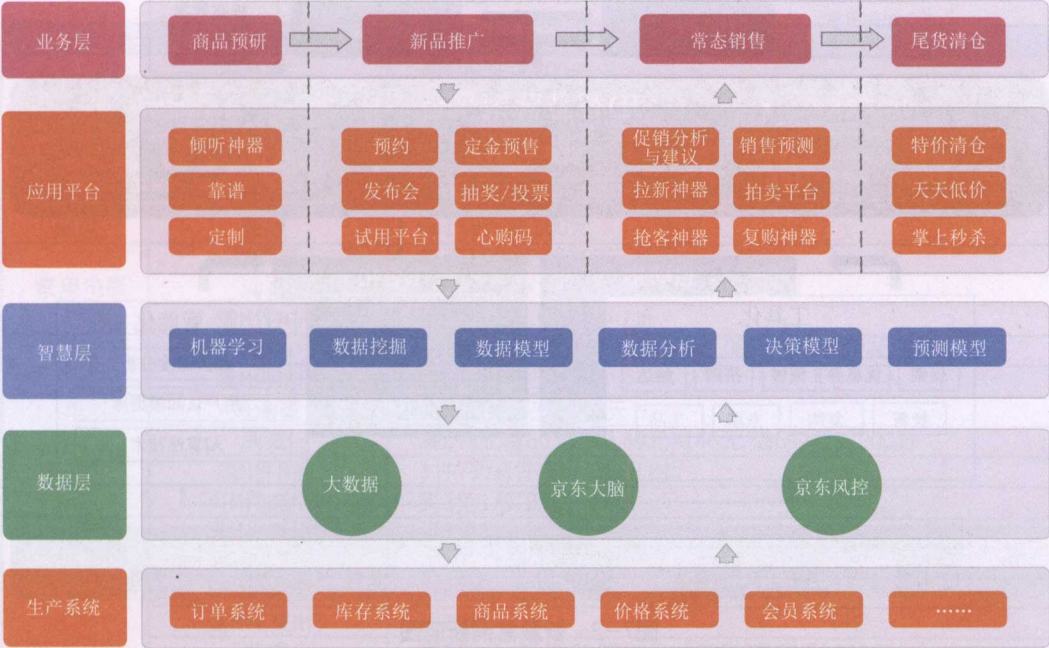


图7-1 智慧营销蓝图初稿

3. 打响推广第一炮

在各产品项目的投入和执行中，智慧营销的方向越来越清晰。交易平台也更坚信智慧营销可以重新定义京东的电商营销。新产品的创建和老产品的智慧化升级齐头并进，使智慧营销平台初露锋芒。

在智慧营销的基础建设日渐成熟后，交易平台启动了智慧营销的推广宣传。在京东内部举办了持续一周的智慧营销周，全面介绍了智慧营销的蓝图规划、产品全貌，并举办了四场分主题演讲。智慧营销周的影响力，引起了业务部门对智慧营销的重视。

智慧营销如火如荼开展起来后，同业务侧的合作从家电延伸到3C，在发展中，又拓展到消费品、时尚等事业部，使各事业部对智慧营销产生认同和信心，联合智慧营销平台对品牌商展开推广和合作。

4. 探索与理念革新

在实践中，交易平台不断对智慧营销进行探索和理念革新，明确了以双环四化（面向用户和商品两个中心环节，提供工具化、系统化、智能化、生态化的解决方案）为核心的新智慧营销蓝图，以及智慧营销的价值主张——提供解决方案加系统工具，输出整个智慧营销平台，如图7-2所示。

智慧营销不只输出解决方案，还通过数据进行深度探查，寻找新的解决方案，提高GMV，促成与品牌方的合作，促进事业部业务目标达成。

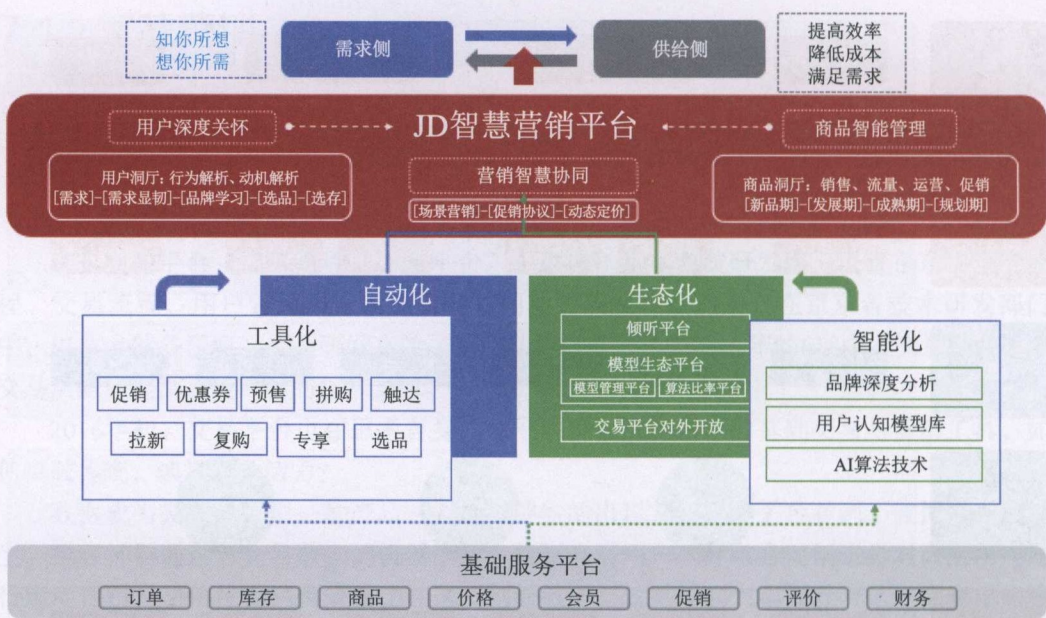


图7-2 智慧营销新蓝图

7.1.2 促销七十二变

每年618是京东人的大日子，如何通过多种多样的促销手段来吸引消费者，是618永恒的主题。在这里回答外界对京东促销的几个疑问。

1. 京东的促销方式有哪些？

京东的促销产品层出不穷，让京东在激烈的市场竞争中立于不败之地。最基本的促销工具包括普通直降、限时抢购、京豆优惠购、套装促销、赠品促销、满减、每满减、阶梯满减、M件减N件、M元买N件、满M件赠N件、满M件N折、满赠、加价购等几十种促销方式。这些促销工具为秒杀、PLUS、闪购、三超（超级品类日、超级品牌日、超级单品日）等营销活动提供了基础产品支撑。另外，我们还有很多互动营销工具，如预约、预售、抽奖、投票、分享有礼等，帮助采销和商家在营销预热阶段积攒人气。

2. 为什么会有这么多的促销方式？

在用户访谈中我们发现，用户总是青睐单品直降或满减这样简单粗暴的降价方式，但是实际情况是各类新奇的促销玩法对消费者总是有意想不到的刺激作用。不断用新的促销玩法来刷新消费者的认知，是营销的重要策略之一。

3. 京东618最喜欢哪种促销方式？

商品按照商品特性和价格带可使用不同的促销方式。如表7-1所示为一些品类的促销类型及优惠力度。每种商品的适用促销类型都不同，以牛奶水饮部门为例，经过多次实验，我们发现当商品是价格低于20元的瓶装水时，做满减促销效果较好，当商品是价格高于20元的整箱装时，做单品直降效果较好。

表7-1 一些品类的促销类型及优惠力度

品类	常见促销类型	常见优惠力度
服饰内衣	满减	满199元减100元
	满M件N折	2件8.8折/8折/9折 3件7折
	买M件减N件	买3免1/买2免1
	M元买N件	99元2件
运动户外	满减	满199元减100元/满159元减20元/满299元减40元
鞋靴	满减	满199元减100元
家用电器	每满减	每满1000元减100元
	满减	满2000元减200元
	满M件N折	2件7.5折/8.5折 3件7折
母婴	满减	满199元减100元/满299元减100元/满199元减50元
	满M件N折	2件8折
礼品箱包	每满减	每满1000元减150元
	满减	满299元减80元/满149元减40元/满699元减200元
	满M件N折	1件7折/2件/8折 3件8.8折
	买M件减N件	买3免1/买2免1
家居家纺	满M件减N件	买3免1
	买M件N折	2件9折/8件 3件8折
食品饮料	买M件N折	2件7.5折/8.5折/8折/7折/5折 3件7件
	满M件减N件	买2免1
	满减	满199元减100元/满99元减20元/满99元减50元
	每满减	每满99元减50元
	加价购	满20元另加9.9元购
电脑、办公	每满减	每满1000元减100元/每满199元减100元/每满199元减80元
	加价购	满50元另加79元购/满15元另加29.9元购 满20元另加39.9元购/满9元另加19.9元购
美妆个护	买M件减N件	买2免1
	满M件N折	2件8折
	满减	满199元减100元/满499元减150元
	每满减	每满99元减50元
医药保健	买M件N折	2件8折/3件7折/4件7折
	满减	满299元减100元/满139元减40元
	每满减	每满99元减15元
手机	满件加价购	满1件加价购
	满赠	满100元赠/满99元赠
	加价购	满15元另加19.9元购
汽车用品	加价购	满200元另加200元购 满20元另加39元购/满19.9元另加9.9元购
数码	加价购	满29元另加24元购/满59元另加19.9元购 满15元另加29.9元购/满20元另加39.9元购
	满减	满199元减40元
家装建材	满件加价购	满1件加价购
	满赠	满599元赠
	满M件N折	3件8折



4. 京东特色促销——跨万店三免一

“跨万店三免一”可以说是京东作为B2C电商的独到优势。京东以自营电商起家，以平台之名招揽天下商家共同为消费者打造一场促销盛宴。一直以来，京东在消费者的认知中，“自营”两个字占主要地位，但是通过促销，消费者对京东的平台化有了全新的认识，知晓京东也有如此多的POP商家和商品，使京东在商品丰富度上一鸣惊人。

“跨万店三免一”打破了购物车按照店铺维度进行商品归堆的逻辑，加入了跨店铺维度，是京东产品创新的一个里程碑。

5. 京东促销氛围——大促预告价

2017年，细心的消费者会发现在618前7天，网站的商品主图图片标签、商详页上出现了大促预告价。

从用户心理上分析，预告价解决了用户大促前购买力减弱、观望大促期间商品价格优惠情况，可预先知道大促期间商品销售价的需求。从采销和商家角度，为618积蓄人气，大幅提升了加入购物车数据，提升了转化率。

7.1.3 优惠券三十六计

在电商营销中，优惠券作为营销利器，是驱动用户和场景链接的最好媒介。利用价格杠杆撬动用户心理的杠杆，广泛利用在促销、引流、用户生命周期和行动激励等场景。尤其在京东，随着业务的不断变化和拓展，优惠券的营销玩法多达500余种，将其作用发挥到极致。从最常见的活动页抢券、会员礼包、购买赠券，到近年新兴的社交化发展的转赠、共享，数据智慧赋能的精准营销，每种玩法的背后都大有玄机。“兵者，诡道也。”618制胜的关键是战术，更是战略。

1. 抛砖引玉，流量锁定

互联网时代的生意，流量即王道，优惠券在流量转化为收益的黄金流程中扮演至关重要的一环。成功的营销是逐层深入的引导、代入感，能一击命中，足以抓住用户的利益点。优惠券本身作为利益点，和其他的促销方式相比，更能够营造一种自己赚到了一笔钱，如果白白过期资产就流失了的心理压迫，所以流量的吸引和锁定效应会更好。

2. 顺手牵羊，关联营销

移动人口红利逐步衰减，粗犷式的引流营销效应也会越来越弱。京东2017年第1季度对外官方公布的活跃用户总量达到2.3亿人，这也就意味着对外的拉新引流的成本越来越高，抓住新用户也越来越难。那么新的品类营销增长点在哪里？京东用优质的服务、低价优质的商品吸引了大量的优质京东粉，但是团队也发现用户在京东的品类消费倾向非常集中，表面看起来是京东粉，其实是京东3C粉或京东家电粉，那么如何利用优惠券做好关联品类的渗透呢？利用多年积累的强势类目带动整体的品类营销发展，也是优惠券的重要突破点。

3. 趁火打劫，高潮期营销

大促最激动人心的是高潮期营销，各种秒杀、优惠券充斥屏幕，白菜价就可以买到心

仪已久的优质好货。高潮期营销的目的总结起来就是：抢流量！抢销量！抢时间！

每天早上10点是京东第一波抢券的时间，这个时候的每秒峰值用户抢券量比平时能达到几十万倍，看似对系统造成巨大压力的不智之举，实际内有乾坤。电商的用户对促销的敏感度普遍较强，当我们抓住日常活跃用户的消费时间规律后，确定了10点开抢的固定场期。用户在限时秒券的过程中会体会到很强的刺激感，而刺激感会让大脑记住这个时间点，从而让用户无论是否有需求都会在10点看优惠。在2017年的618，高潮期被分成多场，10点、14点、18点、20点、24点，每个峰值时刻都可以带动一大批用户回访京东，而复访问的同时，大促为用户推荐的商品池、活动池均会相应的变化，促进用户在一天内多次持续地购买。

限时使用券，是京东2017年618前推出的另一个利用时间砝码促进营销效果的案例。领券时限2h使用，推出后的实验活动效果平均超出普通活动转化率的60%，在部分品类能提高到200%。这背后的秘密就在于，它缩短了用户选购的犹豫期，用高力度的限时优惠帮助用户快速决策，尤其在高客单价品类下，因为没有凑单环节，收效更好。

4.三十六计，唯快不破

如果营销是场战役，那么速度就是制胜的关键。既需要做出及时的玩法变化，顺势而行，也需要快速地做出判断反映。数据化和智能化赋予的营销新生力，精准的品类拉新、复购、品牌高潜在优惠券的玩法应用上面已经不是新课题。传统意义上的精准营销是“精准模型+触达曝光”。但是，用户需求千变万化，触达效率有高有低，速度就是关键。618期间，优惠券的实时精准营销在京东超市拉新战中大放异彩，比传统的精准营销方式提升了极高的效率，关键就在于对进入京东用户的双向实时计算，快速定义用户的精准标签，并在浏览访问结束前，迅速地投放优惠券，抓住用户，完成转化。

此外，实时的效果、实时成本、实时风险都将通过系统告知到业务，并协助运营人员快速做出决策。以秒计数的时代，实时智能才是制胜的未来。

7.1.4 购物助手

电商发展到今天，越来越多的人不再单纯地为商品买单，而是为服务买单。京东长期提供的高品质服务，也越来越多的成为消费者选择京东的原因。购物助手致力于为每一位京东用户提供可信赖的个性化购物服务。利用大数据与人工智能技术，购物助手洞悉用户的购物意向；结合对促销、内容的理解，购物助手可以帮助用户降低购物成本；在用户想到时，甚至在想到之前，购物助手给用户推荐其想要的商品，并提供便捷的免密支付、自动代下单服务，为用户创造极致的电商购物体验。

购物助手计算用户的品类、品牌、商品偏好，进行优质促销资源的定向分发，在避免促销资源浪费的同时，提升了京东的品牌美誉度。

1.洞悉用户购物意向

◆ **购物车数据模型**。相信很多用户有这样的经历，临近618，会把心仪的商品早早地加入购物车，等待合适的出手机会。购物车数据模型的出现，将为这些用户带来福音。购



购物车数据模型分析商品加入购物车的数据，加工处理后开放给采销使用。采销可以根据商品热度等综合因素，在618前放价，为这些用户提供专属的促销优惠。

◆ **β数据模型和复购数据模型。**当然，购物车数据模型只是购物助手利用数据提升用户体验的一个简单尝试。事实上，根据京东的经验，要准确预测用户的购物意向，很难用一个统一的数据模型实现，而是需要扎根商品分类，结合商品特性进行深耕细作。例如，冰箱、空调之类的高客单价耐用品和奶粉、尿不湿等快速消费品之间，几无共性可言。

对于用户购物决策周期比较长的高客单价耐用品，β数据模型将用户购物决策周期细分为不同的阶段：需求显现、品类学习、初级选品、精细选品、选店购买，以针对不同阶段的用户，提供差异化的服务。在品类学习阶段，可以推送一些内容服务，例如“买什么”的商品专辑、达人文章等，帮助用户了解品类的特性；在初级选品阶段，可以推送超级品类日专题活动、优惠券等，为用户介绍品类的优惠活动；在精细选品、选店购买阶段，用户已经锁定了要购买的商品，就需要为用户定制专属的促销优惠，体现京东的价格优势，帮助用户省钱。

对于快速消费品，复购数据模型精确计算用户的购买周期。例如尿不湿，可以精确到用户使用什么样的型号，一天用几片，什么时候用完。在复购商品用完之前，我们为用户提供推送服务，提醒用户及时购买。

◆ **实时数据模型。**上述数据模型采用离线数据进行计算，一般是T+1天产出结果。618大促期间，用户如果没有找到合适的优惠，就有可能失望而去。实时数据模型则是监测用户在京东的行为，实时计算出用户的品类、品牌、商品偏好，并将结果同步给采销。采销就可以根据情报，及时地将用户需要的促销优惠发放给用户。

2.降低用户购物成本

◆ **京东专享。**整合促销资源，帮助用户降低购物成本。京东专享是服务于数据模型的智能促销中台。一方面，京东专享和上游的数据模型耦合，继承了数据模型对用户与商品，商品与商品之间关系的理解；另一方面，京东专享整合了单品直降、优惠券、赠品、京豆、白条免息等可以定向发放给指定用户的专属促销资源，以及秒杀等用户容易错过的非专属促销资源。综合这两方面的能力，京东专享为用户推荐适合的商品，匹配适合的促销，帮助用户省钱。

◆ **聚喜欢**整合内容资源，帮助用户降低时间成本。聚喜欢是一个为用户提供全方位内容服务的系统。对尚在品类学习阶段的用户，聚喜欢为用户提供意向商品相关的内容服务，如商品专辑、晒单、问答、测评、试用报告、达人文章、视频购、直播等，以减少用户的学习成本，节省用户的时间。

3.用户心之所想，即刻所至

◆ **场景1。**用户在浏览京东商品详情页时，专享平台综合用户画像和用户实时行为，若判断用户是忠实用户，直接在页面上提醒用户，可以以专属的优惠价格成交。

◆ **场景2。**用户在京东站外浏览一个小说网站的时候，可以看到一个横幅广告，提醒其在京东常购的某商品降价了，可以去京东看看。

◆ **场景3。**用户打开家里的京东智能冰箱，显示屏上提醒用户某食物快用完了，现在购买比上次购买便宜X元，是否确认购买，用户一键确认后即下单成功。

触达平台整合了短信、邮件、京东商城App消息推送、微信消息推送、广告联盟、JIMI智能机器人、叮咚智能音箱、京东智能冰箱、客服热线等和用户交互的方式。

4. 购物助手未来

购物助手在京东集团2017年度“京芽杯”创新大赛中荣获新芽奖。正如奖项名称的描述，购物助手目前还处于萌芽阶段。要想真正成为用户信赖的购物助手，依赖于数据模型算法的深耕细作，在数据模型、促销、内容、触达等资源的整合上，还有很长的一段路要走。

7.1.5 智慧营销生态化

京东交易平台着眼智慧未来，将用户购物需求和商家销售目标紧密联系，全面构建智慧营销的生态系统。在用户端，从需求产生、品类学习、商品比较、选店下单、售后评价5个阶段进行分析，借助一系列智慧营销产品，向用户提供定制化的管家式服务，实现全流程的用户关怀。在商家端，基于用户行为探查和诉求分析的结果，向商家提供目标客群的真实需求，商家可以及时做出最精准的营销方案，并通过最合理的方式触达目标用户，由此打造出商家与用户双赢的局面。该生态系统依托两大模型效率工具构建：营销数据模型管理平台、京东算法大赛平台。下面重点介绍营销数据模型管理平台。

◆ **系统协作——沟通整合业务需求。**京东营销数据模型管理平台（Model Management Platform, MMP）是京东商城研发部交易平台研发的营销数据分析与管理系统。通过对上游各数据团队研发的营销模型进行收集、整合、分析等一系列管理流程，有效地监测模型优化的效果，甄选最优数据源，为下游所有交易系统主导的品牌竞争力、拉新、复购、抢客等业务系统提供统一、高质量、符合商业预期的数据供给源。目标为实现多数据源商用过程中的风险控制与商业价值最大化，帮助京东营销团队实现各品类GMV的总拉升，同时为数据模型研发团队实现绩效评测与管理。

MMP以模型效果分析为导向，以模型集为类目进行模型管理。对在每个目标产品的模型集内汇总上游各数据团队研发的对应品类营销模型后，对各模型效果进行监控与分析，可实现模型效果的分时段呈现。在同一模型集内，按效果顺序对模型进行组合形成模型组，进而通过建立商用计划，将选定的模型组推送至业务系统，记录模型商用后的效果情况，并分别展示模型、模型组、模型团队整体的效果。

◆ **模型管理——从散兵游勇到精兵强将。**MMP的核心功能之一是模型管理，通过模型集管理、模型管理、模型组管理，来实现对上游数据团队零散的模型进行整合与强化。

首先，搭建散兵集中营——模型集。模型集是对应在一个应用场景中为实现某类特定建模需求而设立的模型内容范围框架，通常在该场景建模前新建，将各数据模型团队研发完成后的模型纳入其中并进行统一管理，对散兵游勇式的数据模型按品类进行归整，集合类似功能模型作统一管理与产出控制。

然后，成立专业作战队——模型。模型是由各数据模型团队根据某一特定营销功能需



求研发的数据营销工具，供对应营销业务系统使用，通常在模型研发完成后，在所属应用场景的模型集内建立。在同一作战集中营（模型集）内，以单个模型为作战单元开展对应目标（商品）的作战活动。即模型是其专属研发团队的成果标志，又是其作用于对应品类营销业务所拉升GMV的产出统计口径。

最后，多队联合协同作战——模型组。模型组是模型集内不同模型的组合，组合方式通常根据特定商用计划需要手动设定，商用计划可根据单个模型产出数据进行，亦可协同多个模型的综合产出数据开展。即为满足不同战役（商用计划）的需要，将单个或多个作战队联合形成技能互补的联盟军团（模型组），形成合力进行团战。在模型集内可设置多个模型组供不同商用计划调用，但同一商用计划只能选用其中一个模型组进行运营活动。

◆ **模型商用——是运输队更是质检组。**MMP核心功能之二是模型商用，通过为不同品类建立特定商用计划，来通道式地实现不同模型集内模型的商用效果。商用计划是将数据团队研发出的模型应用于实际场景中对应业务营销活动，并对模型应用进行过程监控、效果分析的过程，将特定品类商品的模型集、模型、模型组均建立完成后，可为该模型组建立商用计划。


商用计划在流程上是将特定品类商品模型的预测数据（PIN包）由MMP推送至对应业务系统，并获取业务系统对PIN包中的人群开展针对性营销工作后效果的过程，即对于每场战役，MMP把多个作战队技能资源进行整合（对PIN包合并、去重）后，将形成的联盟军团（模型组）输送至前线战场（数据推送至下游业务系统）进行针对性作战（用户营销干预），承担起精细化运输队的职能。

效果分析是对MMP模型管理功能产出价值的评估，用户可通过效果分析功能，直观掌握模型预测效果、模型组预测效果、数据模型研发团队绩效等情况，并以周、月等时间段为单位观察模型产出，即在每场战役后，MMP对作战成果（商用效果）进行实时验收，验收工作包括：对联盟军团总体战役成果的质检（收单GMV、拉升GMV等）、对不同作战小队战绩的质检（准确率、召回率等）、对不同集中营练兵效果的质检（模型团队总贡献）等。具体效果分析主要包括模型商用前和商用后两大模块：

商用前模型效果分析包括模型效果展示、分时间段模型效果汇总、模型对比等功能。MMP在模型集详情页面直观呈现各模型五日前预测用户的自然转化效果情况，包括分数、高潜人数、品类准确率、品牌准确率、SKU准确率、SPU准确率、品类召回率、品牌召回率、SKU召回率、SPU召回率等指标。在分时间段效果呈现页面，展示出模型集内不同模型按照本周平均、上周平均、周环比、本周最高、本周最低、本月平均、上月平均、月环比、本月最高、本月最低等时间段指标统计的效果值。

商用后的模型效果分析包括执行效果、活动效果、排行榜等功能模块。在商用计划列表页面制定商用计划后可选择效果查询时间段，查看模型集内各模型商用后的具体执行效果；同时可在“活动看板”页面查看不同模型执行效果比较，通过时间段、对比模型和数据指标筛选，由呈现出的折线图对比不同模型的执行效果情况。除以上常规运营结果展示外，MMP为采销总监提供特定的效果展示页面，便于业务方管理者清晰、直观地查看相应效果，页面呈现内容包括数据总览、品牌表现、商品表现、采销表现等指标。此外，通过商用排行榜功能，可按团队、按商业计划查看模型总体排行，MMP汇总计算模型在拉升

GMV、收单GMV、风险、转化率、ROI等指标上的得分情况，并进行综合排名呈现。



Ricky: 交易平台数据专家，负责交易平台智慧营销产品规划。

7.2 智能广告

2014年起，京东广告就在一路狂奔。由于发力较晚，创建之初欠下了许多技术债。一边是业务在不停地扩张，一边是系统在不停地迭代。对整个研发的要求非常高，这就像要求一个人不停地吃东西，还得要求不能患有肥胖症。整个广告系统就是在这样的压力下，完成了单业务模式到多业务复合模式的转换，从单系统工作到多系统协作的重构，从人工决策比重过高到业务流程自动化、人工智能决策的升级。如图7-3所示为广告核心系统架构简图，其承担了搜索、推荐、程序化购买等站内外流量的广告，同时支持CPC、CPM、CPS、CPD等丰富多样的广告计费类型。

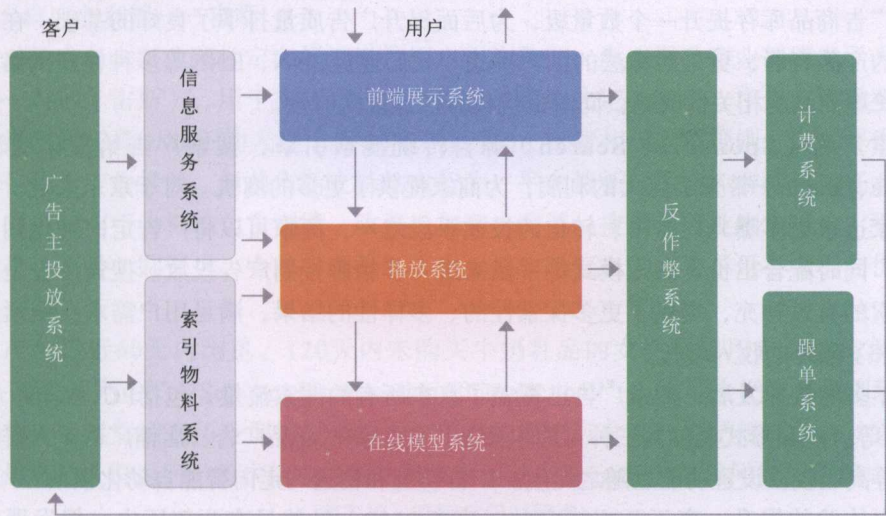


图7-3 广告核心系统架构简图

7.2.1 原生广告

京东展示广告发展这几年，从单纯的CPD广告发展到原生广告、海投计划、秒杀计划等多种新颖智能的广告产品。应用场景以及广告产品的变化促使系统架构以及人工智能算法不断地演进。同时，系统架构与触发排序算法的进步也为广告效果带来质的飞跃。

2014年初，我们接入的广告位主要是网站一些侧边栏广告位，在广告定向上采用最原始的用户行为定向，即根据用户历史浏览、购买、搜索等信息定向。排序采用传统的线性模型预估广告点击率，根据广告点击率排序展现。2015年，我们逐步接入PC、App、微信手Q、M站上的各个推荐位，在推荐结果中插入高质量广告，即原生广告。



我们在保证用户体验的前提下个性化地推荐广告，原有定向方式和排序算法已经不能很好地满足广告产品与用户体验的要求，因此在定向方面引入了DMP定向方式，提升精准定向效果。利用用户基本信息、浏览购买信息等挖掘得到用户性别、人群标签、类目品牌偏好等，构建用户画像。这样，商家可以根据用户画像精准地定向目标用户。在排序方面，升级优化排序模型。使用多个线性模型，分别建模点击率与点击转化率，然后设计了一个融合了点击率与点击转化率的排序公式，点击率和转化率均有明显提升。该建模方式由于是通过人工经验调整融合参数，模型更新上线难度大，点击与订单GMV之间的收益比例不容易平衡。为解决该问题，在后续优化中我们重新建模了一个“展现下单率”模型，并使用DNN深度网络训练。新模型既考虑了点击的因素又顾及到了点击后的下单。使用一个排序模型代替前面的多个模型，大大降低了上线人工调参成本。同时，随着模型的升级，模型参数由千万级变成了数十亿级。

2017年，为了能够大幅简化供应商、商家、采销的广告投放人力成本，以及为广告主提供一个持续高ROI的流量池，我们基于人工智能技术搭建出一套智能投放系统——海投计划。让广告主可以一键式智能投放广告，自动精准定向人群，覆盖到大量的长尾流量。同时将广告商品库存提升一个数量级，为后面提升广告质量打下了良好的基础。在这个简单易用的产品背后，更是对算法的巨大挑战。我们通过CNN、RNN等多种神经网络建模，解决了长尾商品的相关性问题，同时也优化算法保证实时性。

搜索广告（Sponsored Search）源自传统搜索引擎，搜索广告给搜索引擎公司（Google、Baidu）带来了巨大的利润，为商家提供了更多的商机。对于京东来说，搜索广告是商家追求更多曝光、点击、转化的投放手段之一，商家可以将广告定向细化到关键词的粒度，同时配合出价、匹配模式等手段来进一步精确控制广告投放。搜索广告是对自然结果搜索的有效补充，提供了更多探索性的、多样性的结果，满足用户需求；搜索广告给公司带来了额外的收入增长。

三年多的快速发展，搜索广告已覆盖了京东所有的搜索流量，包括PC、App、微信手Q、M站等。产品形式延伸到三种：原生商品广告、侧栏商品广告、店铺广告。人群溢价、抢排名等高级投放设置功能也随之诞生。广告初选和精选均走向智能化预估。

用户体验的提升，离不开智能定向。京东DMP，虽然只有两年历史，但发展甚是迅速，已成为提升体验的一把利剑，除了提供数据管理、人群管理、曝光人数预估、人群画像等基础功能外，还向广告主提供一些高阶个性化功能，如Lookalike（人群扩展）、自助人群挖掘工具、智能定向、场景人群推荐等功能。

1.Lookalike

通常来说，一些大的广告主拥有自己的核心用户，他们有强烈触达与已有核心用户相似的人群的意愿。在这种背景下，为了更好地满足广告主定向需求，平台提出了“新客推荐”定向方式。利用广告主的精准客户数据以及平台拥有的其他用户行为数据，通过“以人找人”的方式进行相似人群扩展，即Lookalike技术，如图7-4所示。满足广告主在获得更大曝光的情况下，达到比使用一些通用标签更好的定向效果。

Lookalike技术的实现方案有很多，本质上是对用户做聚类或者二分类，在工业实现

上，广告系统还要兼顾海量广告单元的扩展效果和效率问题。具体用聚类方法还是二分类方法可以根据应用场景适当选择。目前，精准通中是利用二分类方法选取扩展人群，这种方法可以灵活地控制扩展人群规模，对曝光量和广告效果进行灵活的折中。

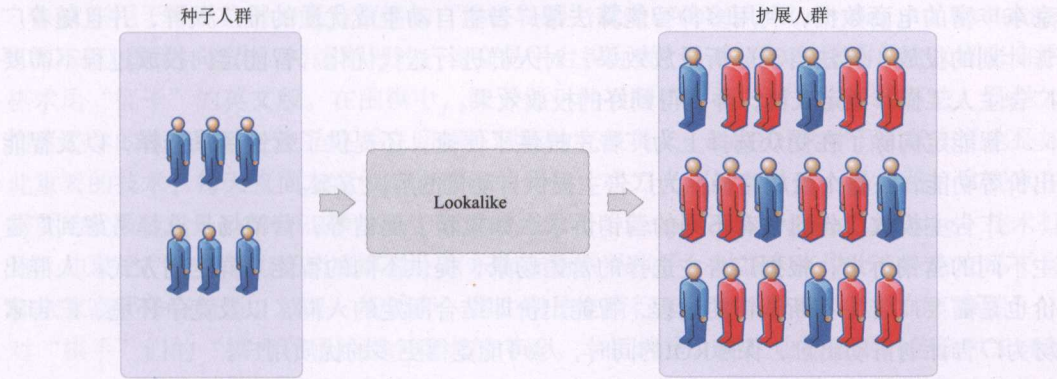


图7-4 Lookalike示意图

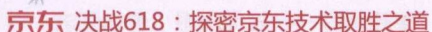
2.人群定向利器——宙斯

通过整合各大事业部的定向标签以及用户行为数据，商业提升事业部研发了人群挖掘引擎——Zeus（宙斯），用于支持广告主个性化、精细化定向需求。广告主只需登录京准通，选择平台工具DMP下的人群挖掘功能，在页面上配置相应人群规则，即可产生精准人群定向包用于广告投放。目前宙斯支持站内行为、平台特征与垂直人群标签的任意交并差组合，其中站内行为标签包含浏览、收藏、加购、购买、搜索词以及广告行为；平台特征包含性别、年龄、购买力、会员等级等20多个标签，垂直人群包含了母婴人群与优惠券人群标签。

以产生最近60天内浏览、120天内未购买牛奶乳品的女性母婴用户人群为例（见表7-2），配置规则为“A Subtract Bintersection Einteresection F”。宙斯会把这些规则翻译成一颗查询树，通过后续遍历查询树以及根据该节点的逻辑组合来产生子规则人群，直到根节点产生最终的人群返回。每次生成访问一节点需启动个Spark任务来访问底层数据。

表7-2 简写字母对照表

定义值		含 义
用户行为	A	最近 60 天浏览了牛奶乳品的用户
	B	最近 120天购买了牛奶乳品的用户
	E	女性用户
垂直人群	F	母婴人群用户
逻辑组合	Union	并集
	Subtract	差集
	Intersection	交集

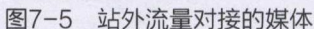


精准通有多维度的受众定向体系，支持广告主各种类型的推广需求，但同样面临着广告主定向选择困惑、学习成本高的诟病。基于这个诉求，我们研发了智能定向系统。依托京东丰富的电商数据，利用多种智能算法帮广告主自动生成优质的推广人群，并且随着广告计划的投放，算法自动分析投放效果，对人群进行迭代优化。智能定向投放过程不需要广告主人工操作圈定人群，并能得到好的投放效果。

广告主投放广告通常有不同的营销诉求，如拉新、促销等，营销场景选择考虑到广告主不同的营销诉求，根据广告主选择的营销场景，提供不同的智能人群定制方式。人群出价也是需要广告主不断试错的过程，智能出价即结合圈定的人群，以及竞争环境、广告素材为广告计划自动出价，保障ROI的同时，尽可能竞得更多的优质用户。

京准通DMP结合日常销售、拉新、转化等多种营销场景，为商家推荐专属定制化人群，满足商家多样化的投放需求。例如，在618期间，针对商家类目为其配置对应的618定制人群，为女性消费场景推荐专门的女性标签等。

相比于寸土寸金的站内流量，站外流量更像是一片蓝海，有形式多样的媒体供我们选择，有更多创新的广告形式供我们尝试。同时，区别于闭环的、用户购物意图明显的站内流量，站外流量的主要目的是品牌宣传和为商城引流，也是京东和友商广告部门正面交锋的主要战场。如何在海量流量中去粗取精、去伪存真，在保证投入产出比的情况下更好地宣传京东的品牌调性、吸引更多的用户来京东逛逛，是站外广告的主要课题。京东的站外广告主要分为直投和程序化购买两种方式。如图7-5所示，勾画了我们近几年在站外开拓出的版图。



276

线的转型，不仅包括今日头条、腾讯新闻等主流媒体，也涵盖了各类垂直媒体以及中长尾媒体。实时竞价流量从0增长到了百亿量级。更加宝贵的是，程序化购买可以做到伸缩自如，在大促期间可以加大引流，在销售淡季可以节约成本，是广告部门的如意金箍棒。

京东用于操纵这根如意金箍棒的“棒法秘籍”就是LADDER（Learning Ad Exchange biDDER）模型，意为“持续学习的广告交易平台投标者”。LADDER缩写的含义则来自围棋术语“征子”的英文版。在围棋中，征子是一种不断“追杀”对方棋子的下法，既是入门者必学的基础技巧，也是高手之间相互制衡的究极奥义。LADDER对于京东来说也是如此重要的技术，每天直面友商们广告部门的竞争，在刀光剑影中杀出一条血路，拿到京东需要的优质流量。LADDER模型基于深度增强学习（Deep Reinforcement Learning）技术打造。如果把广告交易市场当成棋盘，京东和友商就是棋手，对京东来说，在广告交易平台和众家高手争夺流量，就是一场博弈，赌注则是电商行业的市场份额，这样的“豪赌”，对“棋手”们的“棋力”显然是个重大的考验。幸运的是，相比业界通用的传统ECPM竞价算法，LADDER是一个更出色的棋手，在2017年刚一上线，就为当年618期间的站外收入和广告主ROI带来显著提升。广告交易市场的博弈要比棋盘上的对抗复杂得多，也要激烈得多，对任何广告算法来说，收入和ROI同时大幅提升都绝非易事，而LADDER成功地做到了这一点。

7.2.3 PINO人工智能

PINO是一个综合型大规模分布式人工智能平台，是商业背景中的机器学习平台从刀耕火种到集群化自动化演进的产物。以统一框架的思路，用于广告在各类复杂场景的训练和预估，由其支撑的广告业务占比已达70%，为商业提升做出了巨大贡献。该平台负责了机器学习从数据源头到线上服务的全过程。它目前包含模型数据系统、训练系统、评估系统、预测系统，如图7-6所示。

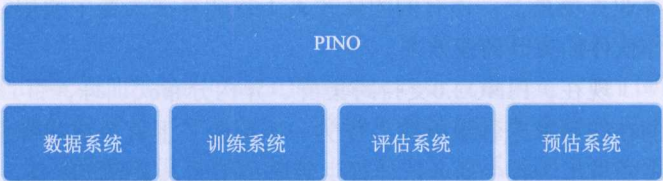


图7-6 PINO四大核心系统

1.模型算法：模型参数从千万级到数十亿级的跨越

我们不妨回看这几年的发展。如图7-7所示，2014年京东广告业务仍处在起步阶段，这时AI支撑的广告业务占比不到10%。2015年，广告流量开始迅速增长，海量数据、复杂模型、实时服务的挑战随之出现。我们当即启动PINO 1.0项目。当年，它支持的广告业务已超过30%。2016年初，团队在复杂模型结构的探索道路上又迈出了坚实的一步，首先将宽深网络引入到推荐场景的模型中，与后来的GoogleWide&Deep模型结构思路不谋而合。随后实现了模型参数从千万级别到数十亿级的提升，且支持了百亿级样本数据，至此由它驱动的广告业务占比已达到70%。2017年，PINO 2.0在大规模分布式的道路上更进一步，在计算性能上比PINO1.0获得了近6倍的进步，在模型类型和模型应用场景上也迎来了爆发期。

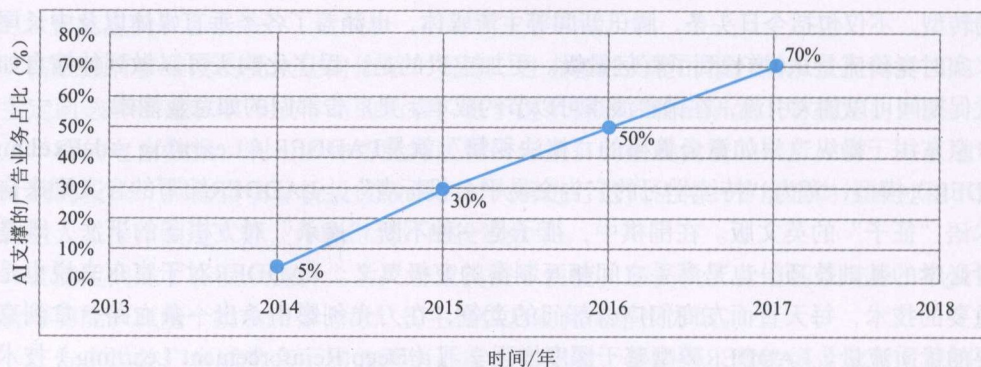


图7-7 人工智能支持的广告业务占比

2.在线服务：海量并发架构演进

不同于传统的人工智能项目，广告、推荐业务不仅需要对复杂问题建模和高效训练，而且需要在很短的时间内对每一次广告、推荐请求做出毫秒级预估服务。目前平台不仅实现了稳定的实时数据流、大规模分布式训练、高度可配置化的模型效果评估、高效的在线实时预估，而且在一步步演进的过程中，通过自身弹性缩放能力，应对了像618大型电商活动的高并发、大吞吐量场景。2017年618期间，它面临了苛刻的考验，经受零点非线性峰值增长的压力，借助公司强大的基础设施，通过弹性扩展，成功应对了流量洪峰。

3.服务场景：从广告业务到全业务

PINO起源于广告、推荐业务，而后发展成支持各类复杂场景的通用算法平台。它覆盖了京东广告的各个业务线，并且推广应用到金融风控、人机对话、销售与物流、图像识别等业务场景上，还有的团队在人工智能算法大赛中使用它崭露头角。它正在帮助京东的合作伙伴们提升商业效率。

现在，PINO2.0支持跨集群、异构分布式训练，百TB级数据小时级处理，数十万并发的在线请求毫秒级响应。作为一个历经了上百个成熟商业应用打磨后的平台，它将会在更多领域里为各个商业伙伴提供优质易用的人工智能服务，助力商业提升！

7.2.4 反作弊系统

广告业务有巨大的商业利益，打击寄生在其中的灰色产业，保护广告主利益，是平台必须认真面对的课题。联盟网站为广告平台带来流量的同时赚取佣金分成，故其有强烈的动机制造假流量。反作弊系统的任务之一就是识别出这些作弊联盟，避免这些联盟上的佣金损失。2015年，京东反作弊系统大规模识别出数十家作弊媒体。现在，每个月系统都会识别出一些作弊网站，我们会将其从联盟中剔除出去。

广告主的广告预算，会因为恶性竞争而遭受恶意点击的消耗，为此我们构建了反作弊系统，识别作弊流量。系统从2014年上线，其间不断进化，识别作弊能力也越来越强，相当一部分流量会被该系统识别为作弊流量。去除这些流量，会保证广告平台的安全性，创造公平的广告环境。在电商平台中，团队除了通过技术手段甄别虚假流量，还会考察流

量带来的订单转化效果，这样作弊的门槛就更高了。但作弊的世界永远充斥着暗流，假流量配合上订单，让这个世界看上去很美好。这些订单有时候是虚假订单，是不良商家通过联盟渠道进行刷单，也可能是就是普通用户的真实订单，用户的订单被联盟通过劫持等手段关联到了虚假流量上。我们构建了劫持订单识别系统、反刷单系统，打击此类作弊。

1. 黑产调研

了解作弊，才能更好地做反作弊，黑产调研是提升反作弊能力的重要手段。流量滋润着整个互联网，在流量大动脉上有勤奋的蜜蜂，也有贪婪的蚂蟥。流量黑产分工明确，目标向钱，在这个圈子里，有流量资源的称为量主，或者出量人。量主的流量来源主要分为两种：机器量，就是通过软件设备模拟人的广告体验行为进行作弊；真实量，就是通过技术手段控制自然流量为黑产端谋取利益。量主的身份小到黑客、流量工作室，大到广告公司和电信运营商多如牛毛，这些人在漫长的流量传输通道的任意节点上都能够获得可观的流量资源。当然，也有直接伪造流量的专业技术团队提供软件或者脚本程序直接牟利。需要流量的我们称为金主，包括广告主、联盟、渠道、站长甚至各级中介。金主对流量的需求主要分为两种：快速获利，采购流量用来获取广告佣金；攻击对手，采购流量攻击竞争对手广告。供需双方都清楚这是作弊行为，为了规避风控，作弊与反作弊对抗强度空前激烈。京东目前已建立了一套比较完善的流量反作弊系统，能够提供及时高效的反作弊能力输出，对绝大多数流量作弊进行有效防御。同时，京东也在投入资源尝试主动攻击型反作弊能力建设：第一，针对流量作弊特点，我们已启动了攻防测试项目，在测试环境下对黑产作弊流量进行量化采样分析，不断发现对方新型作弊特征，优化现有策略，做到事前预警，前置防御；第二，针对大型作弊主体，我们深入调研，直接锁定对方作弊证据并实行线下打击。

2. 反作弊的技术之路

京东广告系统每天有海量的点击曝光，为保护广告主的利益，保证流量质量，我们研发了丰富的反作弊算法。这些算法的任务就是将无效流量、作弊流量从整体流量中分离出来，不参与数据计算和广告计费。为了更快地识别作弊流量，尽早为其他系统提供干净流量，首先进行实时过滤。实时过滤算法在用户、广告点位、流量渠道、广告主等各个维度上融入用户的行为特征，通过统计各维度的点击频次、各维度的行为特征等方法，制定了上百种策略，系统可以在1s内完成作弊流量的识别。不过，实时过滤不能修改之前的判定结果，必然会导致系统性漏过滤的存在，同时实时过滤算法受到时间的限制，不能过于复杂，因为需要开发更复杂的算法来识别更多作弊，基于分布式计算的离线反作弊系统因此诞生。离线反作弊算法通过大数据技术从单个点击、广告点位、IP等各个维度挖掘作弊特征，并训练模型，反馈到实时过滤。离线反作弊采用了无监督和有监督相结合的方式，基于Spark我们研发了一种无监督的异常度模型，可以从广告位、IP、用户等维度，对统计特征进行识别异常，对于单个点击、曝光，我们提取了用户上百维的行为特征，用该异常度模型来识别异常流量，最后将分类的流量通过PINO机器学习平台训练深度学习模型用于实时过滤。我们还在研发一套基于时间序列的异常IP识别模型，根据IP的行为序列，提取几



十维特征进行Embedding，然后采用LSTM等模型学习。

为最大限度地挖掘作弊流量，除了研发反作弊算法，我们也建立了完善的流量质量评估体系，包括人工定期流量排查平台和作弊软件流量测试平台。通过评估体系，我们能够随时跟进市场上新的作弊方式，了解新的作弊平台，研发和完善系统的作弊算法，保护广告主的利益。新策略上线时，使用人工评估平台可以抽取评估使用的各种特征、分布图并展现在Web页面上；评估平台会定期随机抽取数据，评估整个反作弊系统（包括离线和实时）的准确率、召回率。

3. 离线+实时，双管齐下

广告流量反作弊系统采用了离线挖掘与实时过滤相结合的方式，如图7-8所示。

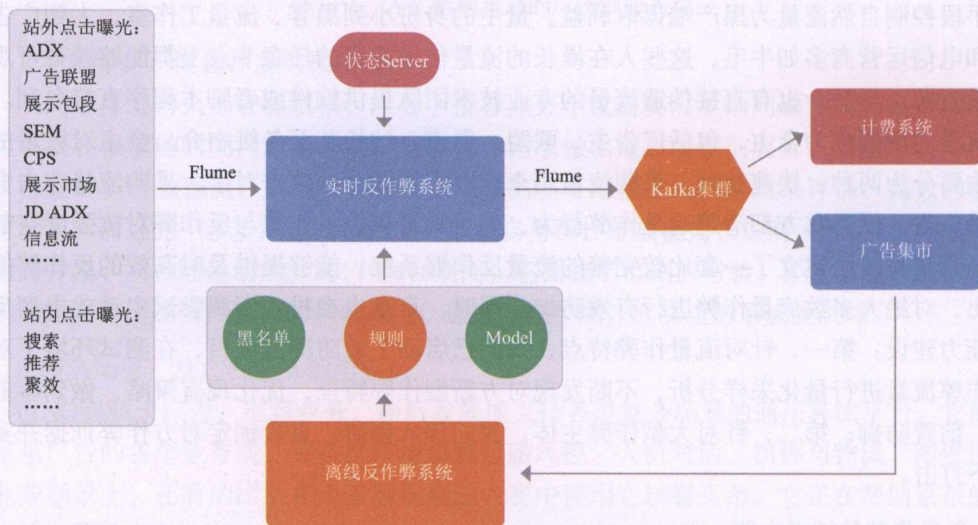


图7-8 反作弊系统架构图

◆ **离线模块。**为了快速处理百亿级日志数据，团队基于Spark、PINO和TensorFlow构建了一套离线反作弊系统。离线系统接入了请求、曝光、点击、行为和订单等多份日志，目前能够通过配置化的方式提取特征，方便算法的调研和模型训练数据的产出。在特征提取的基础上，我们研发了多种无监督和有监督的作弊识别算法，无监督的检测算法能够识别出作弊的媒体、IP和用户等，有监督的机器学习算法（如DNN、LSTM）用来挖掘作弊媒体、IP以及异常流量。

◆ **实时模块。**用户的点击、曝光以Flume流到达实时反作弊模块，基于sofa-pbRPC通信，我们研发了一套支持大规模数据并发处理的实时反作弊系统。其会首先过滤命中黑名单的流量，并能够实时地从多个维度统计频次、行为等特征，将结果缓存在状态服务器中，然后根据上百维的规则库识别无效点击和作弊点击。

离线特征数据丰富，能够有效识别作弊。那么如何将挖掘出来的作弊反馈到实时呢？一种简单的方式是挖掘出黑名单、规则。另外一种更准确有效的方式是，基于前面流量识别部分的异常度模型识别出作弊流量后，使用PINO平台提供的LR+DNN深度神经网络模型学习作弊行为生成模型文件，将其推到在线预估服务上；流量流经反作弊系统时可以

实时请求在线预估服务，判别是否作弊，该方案目前的准确率在95%以上。

广告反作弊系统基于海量的日志数据，采用离线和实时相结合的方式，通过多种机器学习算法，从多个维度、多个角度挖掘机器和人为的刷量、恶意点击，过滤虚假流量，切实保障广告主的利益，并维护广告生态的良性发展。

4. 效果保障

反作弊策略对于准确率的要求非常高，每一次新策略的上线、现有策略的优化都需要经过严格的人工评估。反作弊产品团队有一套标准化的评估流程进行准确率的评估，策略准确率达标才能上线。策略上线前，会进行上线前后作弊数据diff分析，排除特殊场景导致的误杀；上线后，也会持续观察策略效果，保证策略的表现符合预期。

反作弊系统在保证广告主和广告平台的利益上做出了重大贡献。以某国际知名IT公司为例，该公司通过京东广告平台投放站外广告，在使用京东结算数据的同时接入了第三方监测数据。后发现某段时间第三方监测的点击数高出京东结算数据30%以上，经排查，是由于反作弊系统发现了部分作弊流量并进行了过滤，保障了广告主的利益。最终广告主成本得以控制，比以往历史投放效果提升了20%。



阿亮：商业提升事业部质量专家，广告测试、运维、监控、618备战负责人。

大山：商业提升事业部研发总监，广告程序化购买负责人。

7.3 精准搜索

在互联网信息化高度发达的今天，人人都可以拿着智能手机随时随地掌握最新鲜的资讯，消费者对生活品质的追求和服务体验的要求越来越高，这就推动着电商在提高用户体验的道路上不断前进。搜索作为电商主要入口之一，快速准确地提供结果尤其重要。

就京东的搜索场景来说，数据量和用户量一直保持快速增长，再加上业务不断发展创新，对搜索系统提出了越来越高的要求。同时，随着新技术的引入（如强化学习、千人千面），对实时性和召回多样性提出了新的挑战。海量的商品数据，如若搜索精准性能够提高，将使用户在购物时在小范围内就能快速筛选出自己心仪的商品，将会极大地提升购物体验。

京东的业务增长速度惊人，对于后台系统的要求越来越高。2013年的时候，搜索的索引数据还是千万级，到2015年已经是亿级，并且可预见的增长率也是呈直线飞跃。同时，新增用户数量也保持高速增长。在面对如此大量的数据量和用户群的增长背景下，精准搜索的必要性和商业价值凸显，成为技术部门势在必行的工作。

数据的爆发式增长对系统的挑战越来越大，必然会推进系统架构的升级。应对不同的数量级，架构模型是不太一样的。京东搜索系统经过发展，目前已经能够支持数十亿级的商品量，日PV量也达到10亿次以上，并且能够通过线性扩展，支持更大的数据量和用户规模，为提高搜索使用体验发挥了巨大作用。



7.3.1 搜索架构演进

京东搜索引擎是搜索部门自主研发的商品搜索引擎，主要功能是为海量京东用户提供精准、快速的购物体验。目前入口主要有PC端、移动端、微信手Q、移动列表页、店铺搜索、店铺列表等。伴随着业务与数据的增长，京东搜索系统也经历了几代的迭代演进，体系结构越来越复杂，支持的功能也越来越多样化。下面介绍搜索系统的演进历史。经过多年的618检验，系统的并发性与稳定性得到了充分的验证。

京东最初的商品搜索系统（V1版），是基于Lucene搭建的，数据存储使用的是SQL Server数据库，那时是“台哥”一个人扛大旗，自己负责搜索页和列表页。随着公司的快速发展，对搜索相关性、排序提出了更高的要求。为了满足这些需求，需要有一套能够高度定制的搜索系统，2011年11月启动新版搜索架构升级项目。由于京东每年在618举行店庆促销，留给项目的时间只有5个月。当时开发人员只有7个人，需要在如此短的时间内开发上线一套新的架构并且完全兼容现有的业务，同时要能够在618超高流量情况下保持稳定，因此挑战非常大。最终团队花了2个月时间，用C++开发了一套新的搜索服务（V2版）。恰逢图书搜索频道搭建，有众多垂直化需求，新的系统可以更好地提供支持，于是选择图书搜索作为新版架构的第一个产品，2012年2月完成图书搜索上线。经过3个月的图书搜索实验，新系统效果显著，主站的商品搜索随后也切换到V2版，并在618期间承担了10%的流量，新版搜索服务保持高度稳定。因此618后很快将全部流量切换至V2版。

那时商品数据量在百万级，都存储在SQL Server数据库里，通过循环扫库建全量索引即可完成。但随着商品越来越多，更新越来越慢，一次全量更新从最初的几分钟到后来的2h。虽然通过按品类并行索引有所缓解，但没有根本解决时效性问题。尤其是在618零点，大量促销生效，价格批量变化。随后的几分钟，由于很多商品迅速被抢光，库存状态变为无货。这些都无法及时地在搜索中体现，严重影响了用户体验。

为解决时效性问题，2013年开启了搜索实时更新项目，在全量的基础上设计数据动态更新的方案。首先接入了库存、价格等对时效性比较敏感的数据更新。各个上游系统在数据发生变化时，发送消息通知，实时索引模块接收消息后，生产对应的数据推送给线上的索引服务。后来又接入商品上架消息，对新上架的SKU生产增量索引，调用各个业务接口组装实时数据，再采用同样的方式同步到线上，由线上服务器按树状分层更新。实时索引模块同时还会把最新数据写入搜索异构的数据中心，供全量索引生产使用。完成这项改造后，对应的数据能在一两分钟内更新到线上所有服务。V2版搜索系统的架构如图7-9所示。

该版架构中检索服务是集中式的，数据一致性维护成本很低，集群扩容缩容便捷。在应对大促的备战时，通过集群扩容便可保障更高请求压力的实时性和稳定性。经过持续优化，迭代升级，一直良好地支撑着京东业务的飞速发展。直到2016年底，才完成其历史使命，将接力棒交给新一代分布式搜索系统。在它服役的这几年，满足了京东的数据量数十倍的增长，随着业务增长不断迭代，代码量也积累到了十万行的规模。但它也存在局限性，由于数据的飞速增长，单索引数据已经无法完全容纳全部商品数据，在2015年，商品数量早已突破数千万，对现有系统的挑战越来越大。当时的团队成员还只有十几个人，要维护线上系统和持续迭代新业务。在新系统的研发又势在必行的情况下，立项了分布式搜

索引引擎项目（V3版）。

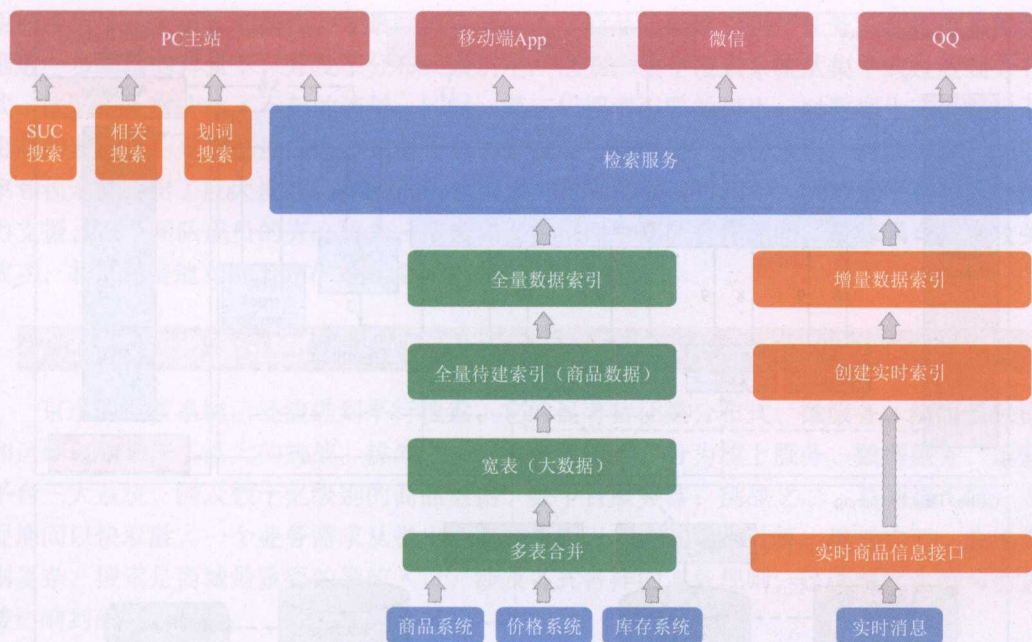


图7-9 搜索系统（V2版）架构

7.3.2 分布式搜索引擎

分布式搜索引擎项目（V3版），京东内部项目名称为平行搜索。设计目标为支持数十亿商品数据量，后续经过一系列迭代升级后，可支持百亿级商品数据量。

分布式引擎在2016年8月开始灰度上线。在2017年元旦前夕全面接管各渠道流量（包括PC主站和京东App移动端），宣告新一代分布式搜索系统的正式服役。并在2017年618中表现出色。上线后价值贡献统计结果为：PC搜索转化率提升1.46%，UV价值提升1.09%；移动端App搜索转化率提升0.71%，UV价值提升0.55%。

同比2倍压力下平均响应延时缩短了一半，并发性、稳定性、实时性都得了实践检验。索引数据的商品量，从2016年初到8月增长了7倍。引擎系统也经历了两次扩容，后来内部催生了一个建立在平行搜索系统之上的具有里程碑式的大项目——索引分层，彻底解决爆发式增长的数据量对搜索引擎提出的挑战。分布式搜索引擎的逻辑结构如图7-10所示。VIP为京东内部负载均衡系统。Root为搜索引擎的接入层，承担流量整合与安全策略的服务。Parent为分布式结果集汇总及搜索业务处理层。Leaf为检索服务，承担索引实时更新与召回任务。LoadBalancer为集群负载均衡器。OnlineUpdating为在线实时更新组件。

平行搜索系统上线后，开始承接更为复杂的业务需求，为丰富的业务场景提供了良好的展示平台。新系统上线不久，就遇到了数据量过快的挑战，通过扩容分片从容应对。整个扩容过程是在线完成的，即新机器部署好后，通过负载均衡服务动态地添加到生产集群，在用户无任何感知的情况下完成线上系统扩容。并且在扩容前后的可用率始终维持在



三个9（99.9%）的水平，动态伸缩的设计得到实践验证。

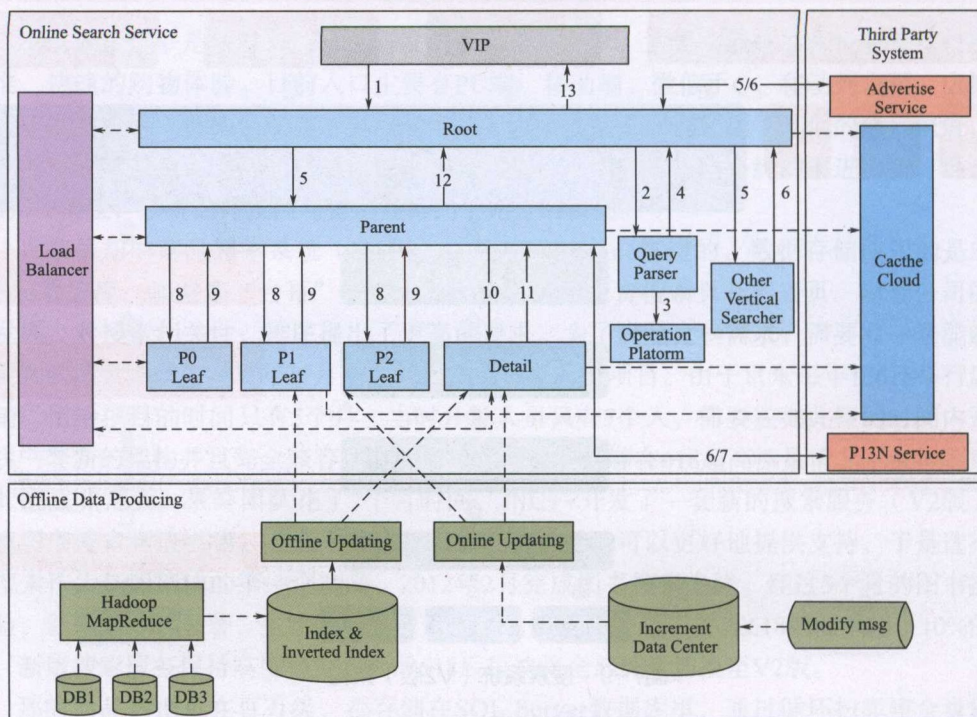


图7-10 分布式搜索引擎的逻辑结构

7.3.3 数据索引

引擎架构的不断迭代，对数据的多样化和实时性也提出了高要求。因此，索引生产是搜索系统一个非常重要的环节。早期的数据生产采用的是集中式索引分布式生产框架，该框架是为了解决集中式索引的生产问题，这个版本后系统中再无MySQL依赖，我们将整个数据生产系统重新抽象成多个处理单元，可以方便地进行业务迭代。对数据系统来说这是一次全新的尝试，该版本索引上线过程非常的坎坷，数据团队一个产品渠道一个产品渠道的推动，好在最后所有的渠道平台都顺利上线，期间没有发生一起紧急回滚的事故，这在大规模索引切换场景下是非常难得的。

面对快速增长的数据量，基于MySQL的索引生产架构在这种大规模数据场景下显得力不从心，况且京东高度复杂而且特有的业务逻辑，使搜索系统急需一种处理大规模数据又能方便处理复杂业务逻辑的索引生产框架，京东Hadoop索引生产框架就在这种大背景下应运而生。在设计Hadoop索引框架之初，由于搜索数据系统以前一直基于MySQL架构，对Hadoop系统没有太多的技术积累，在研发的过程中为了解决技术难题经常加班。在那段时间里，数据组经常工作到很晚，打车回家成了家常便饭，每逢月末报销的时候，每个人都提报厚厚一打的打车发票，这些发票也成了以后数据系统顺利上线的有力见证。

在框架设计之初，考虑到后期迭代的速度以及维护的成本，团队采用了Streaming的

技术方向；为了方便处理复杂的业务逻辑，我们又引入了Hive。在整个索引演进的过程中我们经历了几个版本的迭代，才最终肩负起数十亿商品量的索引生产任务。在搜索系统数据进一步暴增的背景下，开发了分布式索引生产框架，整个搜索系统从集中式过渡到分布式，这又是一次非常了不起的跨越。同时，第三代搜索引擎的诞生，对数据生产也提出了更高的要求。一是数据量的爆炸式增长，二是需要分布式生产多份索引，对数据生产的效率和稳定性提出了巨大挑战。在Hadoop索引系列框架的研发过程中，大数据平台团队的全力支援，以及团队成员的齐心协力，搜索和大数据两个团队合作无间，最终联合技术攻关成功，非常完美地支持了分布式搜索引擎的索引数据更新。

7.3.4 搜索系统的DevOps之路

京东的搜索系统已经演进到平行搜索，它的显著特征是分布式、微服务，给质量保证和运维工作带来了巨大的挑战。挑战之一，系统规模大，分为线上服务、数据服务、运营平台三大系统，涉及数十亿级别的商品数据、数千台服务器；挑战之二，系统迭代快，大促期间以快取胜，一个业务需求从提出到上线时间以小时或天来计算；挑战之三，业务规则复杂，搜索是商城最重要的导流入口，涉及各式各样的业务规则，这些规则生效与否直接影响到商家的利益。

传统测试开发的工作模式是理解业务，编写测试工具，协调沟通。我们认为：这些工作依然重要，但是无法完全解决面临的挑战，解决问题的钥匙应该是DevOps。我们的体会无论是质量保障，还是运维管理都需要平台化、自动化来支撑，一方面实行持续集成和持续交付；另外一方面，让业务、产品经理、研发、算法和运维的同事都参与到测试工作中来。在电商领域，质量保证工作除了线下测试，还需要协调业务、产品经理、研发和运维，把测试延伸到线上，做到线上全业务监控，第一时间发现业务的问题。对于测试本身来说，单纯依靠人力，已经无法给搜索系统设计出足够多的测试用例来覆盖日益变化的业务规则和用户行为了；需要使用大数据处理的方式来支撑我们的测试工作，帮助我们生成测试数据和分析用户行为。

关于以上几点，团队正在尝试用机器学习的方式找出badcase、系统漏洞，支持产品设计和架构演进。另外，团队正在尝试把机器学习引入繁重的日常测试结果分析工作，通过技术把人从某些重复的测试工作中解放出来，从事更有价值的、创造性的工作。

1. 质量保障的平台化自动化

DevOps的本质是在平台化自动化基础上的敏捷。团队正在持续迭代开发一套DevOps平台，它给面向最终用户的线上服务、数据服务和运营平台提供闭环支撑。闭环支撑包括两层含义：从系统研发的生命周期角度来看，从研发的调试环境，到测试环境，再到预发环境，最终到线上环境部署，DevOps平台提供一致的编译方式、配置管理和部署方式，用平台化自动化方式提高研发工作的效率；从自动化运维角度来看，DevOps提供了故障管理、系统管理、变更管理、资源管理和数据管理等一整套解决方案。通过UI和自动化方式，保证运维操作准确度，提高工作效率。



2. 测试从线下延伸到线上

由于平行搜索系统中业务的重要性，除了系统和应用级别的监控，我们添加了大量核心业务的监控。具体实现是通过业务测试脚本不断地验证线上服务，把测试结果写入监控数据库，并通过报表系统以时间序列展示业务状态。我们选用的监控系统架构是 OpenTSDB+HBase+Grafana。监控只能看到业务历史状态，还需要告警系统第一时间把业务失效的信息通过邮件和短信方式告知相关人员，具体实现是在 Icinga 系统上进行二次开发。如图 7-11 所示的实时业务监控是搜索业务监控的几个具体例子，其中全球售、山姆会员是具体业务，每个子图对应业务可用率百分比，可以按 App、PC 等渠道来查看不同渠道下的业务情况。

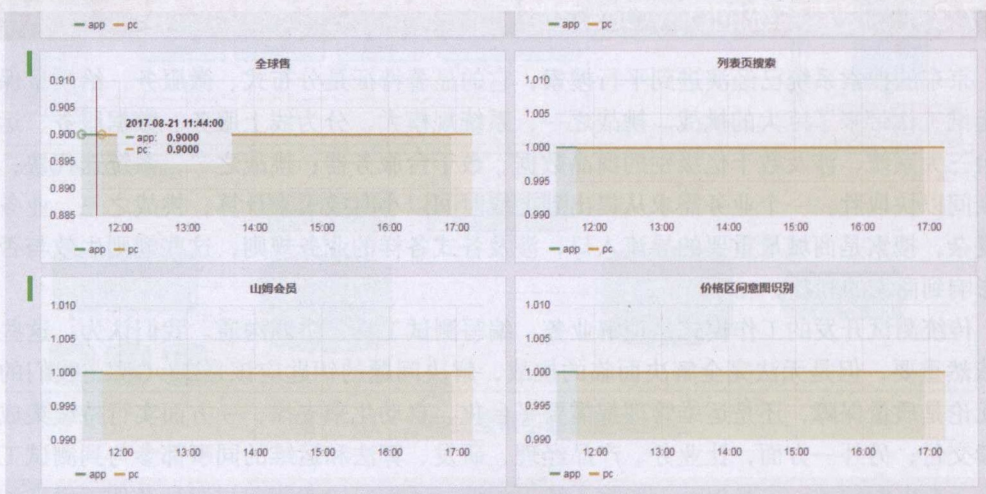


图7-11 实时业务监控

3. 大数据支撑质量管理

平行搜索系统业务涉及数千个三级类目、数十亿个商品和数亿个用户行为，我们把用户行为的大数据进行抽取、清洗、汇总和分析，定期生成测试数据。以最少的测试用例，覆盖更多的功能和用户行为，并且保证了测试用例的实效性。

技术与公司业务是循环驱动的关系，业务发展对技术提出新的挑战。技术进步又促进业务高效发展。京东搜索就是在这样的场景下不断进步，成长为核心系统的。现如今，京东每日有接近一半的交易额来自搜索流量。

对于搜索系统的发展，将会在业务场景的驱动下继续前行，随着数据量和用户量的进一步扩大，对于搜索的实时性、稳定性以及精准性，都会提出更高要求。电商所特有的大数据，应用深度学习等前沿技术持续提高搜索个性化与精准性体验，起到了极大的促进作用。这些都将未来成为搜索系统的发展动力。用技术服务亿万京东用户，持续提升用户购物的幸福感，是京东搜索团队不变的使命。



cnycode：搜索架构师，负责京东新一代分布式搜索引擎架构研发工作。

7.4 智能推荐

在电商领域，推荐的价值在于挖掘用户潜在购买需求，缩短用户到商品的距离，提升用户的购物体验。

京东推荐的演进史是绚丽多彩的。京东的推荐起步于2012年，当时的推荐产品甚至是基于规则匹配做的。整个推荐产品线组合就像一个个松散的原始部落一样，部落与部落之前没有任何工程、算法的交集。2013年，国内大数据时代到来，一方面如果做的事情与大数据不沾边，都显得自己水平不够，另外一方面京东业务在这一年开始飞速发展，所以传统的方式已经跟不上业务的发展了，为此推荐团队专门设计了新的推荐系统。

随着业务的快速发展以及移动互联网的到来，多屏（京东App、京东PC商城、M站、微信手Q等）互通，推荐类型从传统的商品推荐，逐步扩展到其他类型的推荐，如活动、分类、优惠券、楼层、入口图、文章、清单、好货等。个性化推荐业务需求比较强烈，基于大数据和个性化推荐算法，实现向不同用户展示不同内容的效果。为此，团队于2015年底再次升级推荐系统。2016年618期间，个性化推荐大放异彩，特别是团队开创的“智能卖场”，实现了活动会场的个性化分发，不仅带来GMV的明显提升，也大幅降低了人工成本，大大提高了流量效率和用户体验，从而达到商家和用户双赢，此产品获得了2016年度的集团优秀产品。为了更好地支撑多种个性化场景推荐业务，推荐系统一直在迭代优化升级，未来将朝着“满屏皆智能推荐”的方向发展。

7.4.1 推荐产品

用户从产生购买意向，到经历购买决策，直至最后下单的整个过程，在任何一个购物链路上的节点，推荐产品都能在一定程度上帮助用户决策。

1.推荐产品发展过程

推荐产品发展历程主要经历了几个阶段（图7-12），由简单的关联推荐过程到个性化推荐，逐步过渡到场景智能推荐。从相关、相似的产品推荐过渡到多特征、多维度、用户实时行为、结合用户场景进行的全方位智能推荐。



图7-12 推荐产品发展历程



2.多屏多类型产品形态

多类型主要指推荐类型覆盖到多种类型，如商品、活动、分类、优惠券、楼层、入口图、文章、清单、好货等。在移动互联时代，多屏场景非常普遍，整合用户在多屏的信息，能使个性化推荐更精准。多屏整合的背后技术是通过前端埋点，用户行为触发埋点事件，通过点击流系统进行多屏的行为信息收集。这些行为数据通过实时流计算平台来计算用户的兴趣偏好，从而根据用户兴趣偏好对推荐结果进行重排序，达到个性化推荐的效果。京东多屏终端如图7-13所示。



图7-13 京东多屏终端

7.4.2 推荐系统架构

1.整体业务架构

推荐系统的目标是通过全方位的精准数据刻画用户的购买意图，推荐用户有购买意愿的商品，给用户最好的体验，提升下单转化率，增强用户黏性。推荐系统的业务架构如图7-14所示。

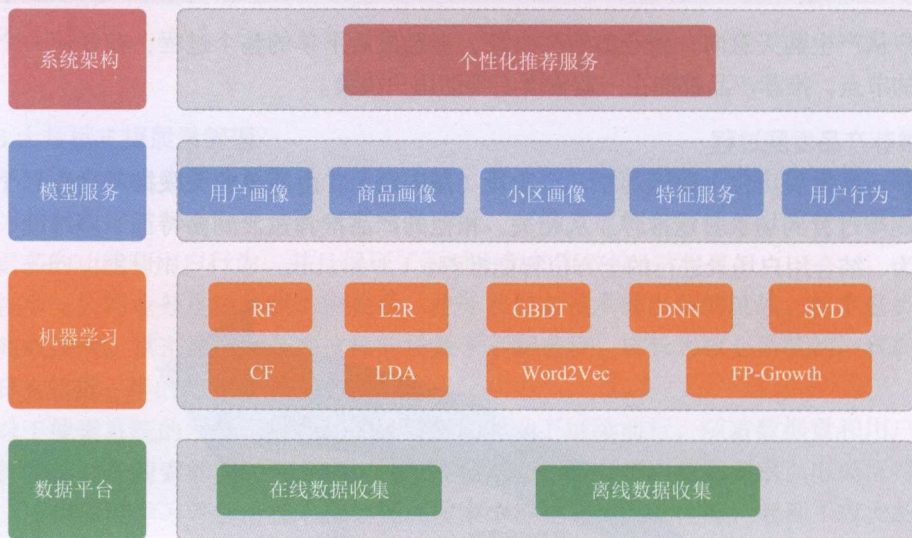


图7-14 推荐系统的业务架构

- ◆ **系统架构。**对外提供统一的HTTP推荐服务，服务京东所有终端的推荐业务。
- ◆ **模型服务。**为了提高个性化的效果而开发的一系列公共的个性化服务，用户维度有用户行为服务和用户画像服务，商品维度有商品画像，地域维度有小区画像，特征维度有

特征服务。通过这些基础服务，让个性化推荐更简单、更精准。

◆ **机器学习**。算法模型训练阶段，尝试多种机器学习模型，结合离线测评和在线A/B，验证不同场景下的算法模型的效果，提高推荐的转化率。

◆ **数据平台**。数据是推荐的源泉，包括数据收集和数据计算。数据虽然是整体推荐架构的最底层，却是非常重要的，因为数据直接关系到推荐的健康发展和效果提升。

2. 个性化推荐架构

在起步初期，推荐产品比较简单，每个推荐产品都是独立服务实现。新版推荐系统是一个系统性工程，其依赖数据、架构、算法、人机交互等环节的有机结合。新版推荐系统的目标，是通过个性化数据挖掘、机器学习等技术，将“千人一面”变为“千人千面”，提高用户忠诚度和用户体验，提高用户购物决策的质量和效率；提高网站交叉销售能力，缩短用户购物路径，提高流量转化率（CVR）。目前新版推荐系统支持多类型个性化推荐，包括商品、店铺、品牌、活动、优惠券、楼层等。新版个性化推荐系统架构如图7-15所示。

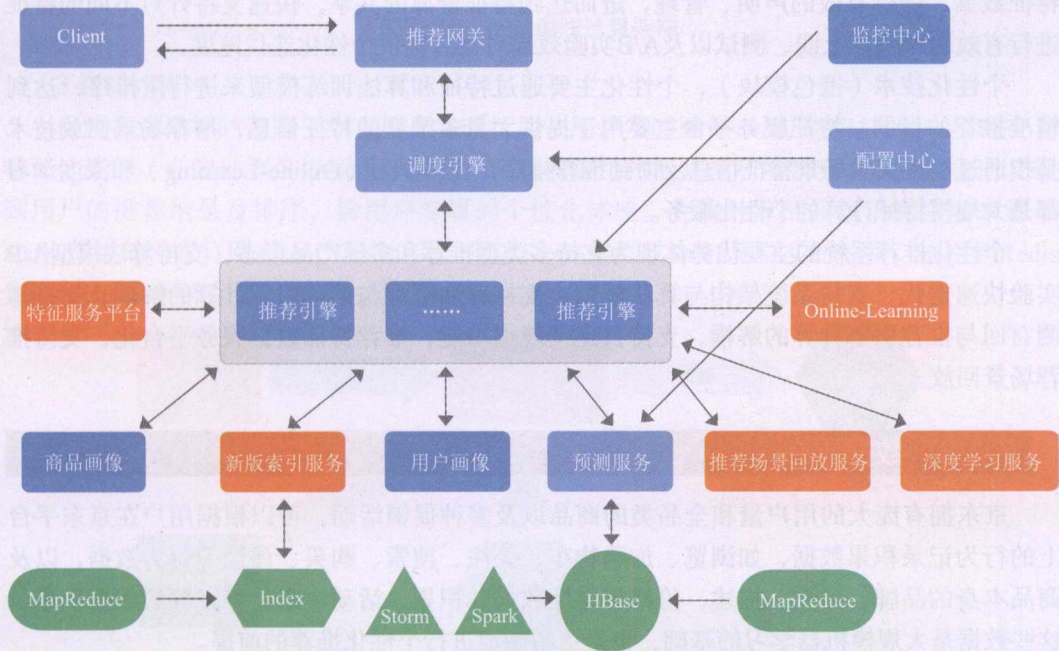


图7-15 新版个性化推荐系统架构

个性化推荐系统架构图中不同的颜色代表不同的业务处理场景：

数据处理部分（最底层绿色模块），包括离线数据预处理、机器学习模型训练，以及在线实时行为的接入、实时特征计算。

推荐平台（蓝色模块），主要体现响应用户请求时推荐系统的各服务模块之间的交互关系。推荐系统核心模块：

◆ **推荐网关**。推荐服务的入口，负责推荐请求的合法性检查、请求分发、在线Debug以及组装请求响应的结果。



◆ **调度引擎**。负责推荐服务按策略调度及流量分发，主要根据配置中心的推荐产品的实验配置策略进行分流，支持按用户分流、随机分流和按关键参数分流。支持自定义埋点，收集实时数据；支持应急预案功能，处理紧急情况，秒级生效。

◆ **推荐引擎**。负责推荐在线算法逻辑实现，主要包括召回、过滤、特征计算、排序、多样化等处理过程。

◆ **个性化基础服务**。目前主要个性化基础服务有用户画像、商品画像、用户行为、预测服务。用户画像包括用户的长期兴趣、短期兴趣、实时兴趣。兴趣主要有性别、品牌偏好、品类偏好、购买力等级、自营偏好、尺码颜色偏好、促销敏感度、家庭情况等。商品画像主要包括商品的产品词、修饰词、品牌词、质量分、价格等级、性别、年龄、标签等。用户行为主要获取用户近期行为，包括用户的搜索、点击、关注、加入购物车、下单等。预测服务主要是基于用户的历史行为，使用机器学习训练模型，用于调整召回候选集的权重。

◆ **特征服务平台**。负责为个性服务提供特征数据和特征计算，特征服务平台主要针对特征数据，进行有效的声明、管理，进而达到特征资源的共享，快速支持针对不同的特征进行有效的声明、上线、测试以及A/B实验效果对比。提供个性化迭代速度。

个性化技术（橙色模块），个性化主要通过特征和算法训练模型来进行重排序，达到精准推荐的目的。特征服务平台主要用于提供大量多维度的特征信息，推荐场景回放技术是指通过用户实时场景特征信息反馈到推荐排序，在线学习（Online-Learning）和深度学习都是大规模特征计算的个性化服务。

个性化推荐系统的主要优势体现为支持多类型推荐和多屏产品形态，支持算法模型A/B实验快速迭代，支持系统架构与算法解耦，支持存储资源与推荐引擎计算的解耦，支持预测召回与推荐引擎计算的解耦，支持自定义埋点功能；推荐特征数据服务平台化，支持推荐场景回放。

7.4.3 数据平台

京东拥有庞大的用户量和全品类的商品以及多种促销活动，可以根据用户在京东平台上的行为记录积累数据，如浏览、加购物车、关注、搜索、购买、评论等行为数据，以及商品本身的品牌、品类、描述、价格等属性数据的积累，活动、素材等资源的数据积累。这些数据是大规模机器学习的基础，也是更精确地进行个性化推荐的前提。

1. 数据收集

用户行为数据收集流程一般是用户在京东平台（京东App、京东PC网站、微信手Q）上相关操作，都会触发埋点请求点击流系统（专门用于收集行为数据的平台系统）。点击流系统接到请求后，进行实时消息发送（用于实时计算业务消费）和落本地日志（用于离线模型计算），定时自动抽取行为日志到大数据平台中心。算法人员在数据集市上通过机器学习训练模型，这些算法模型应用于推荐服务，推荐服务辅助用户决策，进一步影响用户的购物行为，购物行为数据再发送到点击流，从而达到数据收集闭环。

2. 离线计算

目前离线计算平台涉及的计算内容主要有离线模型、离线特征、用户画像、商品画像、用户行为，离线计算主要在Hadoop上运行MapReduce，也有部分在Spark平台上计算，计算的结果通过公共导数工具导入存储库。团队考虑到业务种类繁多、类型复杂以及存储类型多样，开发了插件化导数工具，降低离线数据开发及维护的成本。数据离线计算架构如图7-16所示。

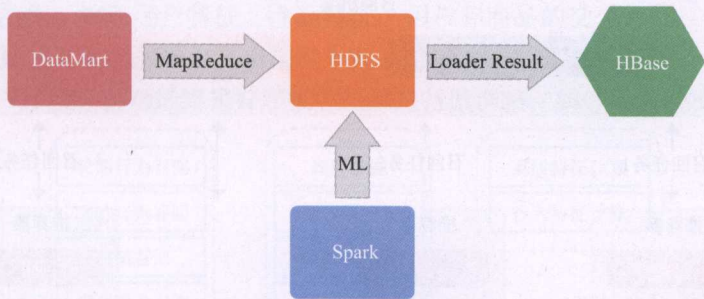


图7-16 数据离线计算架构

3. 在线计算

目前在线计算的范围主要有用户实时行为、用户实时画像、用户实时反馈、实时交互特征计算等。在线计算是根据业务需求，快速捕捉用户的兴趣和场景特征，从而实时反馈到用户的推荐结果及排序，给用户专属的个性化体验。在线计算的实现消息主要来源于Kafka集群的消息订阅和JMQ消息订阅，通过Storm集群或Spark集群实时消费，推送到Redis集群和HBase集群存储。数据在线计算框架如图7-17所示。

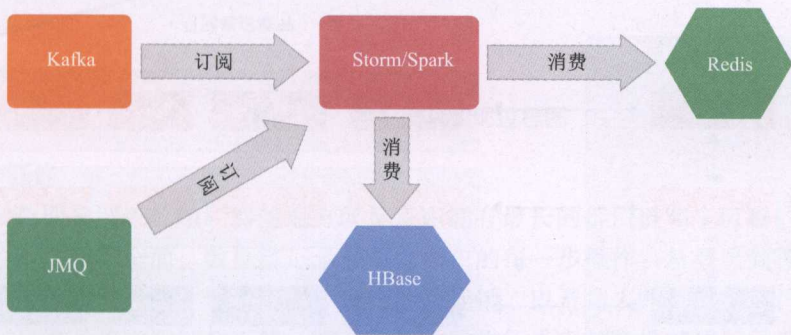


图7-17 数据在线计算架构

7.4.4 关键技术

推荐系统涉及的技术点比较多，考虑到篇幅有限，这里重点介绍个性化推荐中比较重要的部分。

1. 推荐引擎

个性化推荐系统的核心是推荐引擎，推荐引擎的一般处理过程是召回候选集，进行规



则过滤，使用算法模型打分，模型融合排序，推荐结果多样化展示。主要使用的技术是机器学习模型，结合知识图谱，挖掘商品间的关系，按用户场景，通过高维特征计算和海量召回，大规模排序模型，进行个性化推荐，提升排序效果，给用户极致的购物体验。

推荐引擎处理逻辑主要包括分配任务，执行推荐器，合并召回结果。推荐器负责召回候选集、业务规则过滤、特征计算、排序等处理。推荐引擎技术架构如图7-18所示。

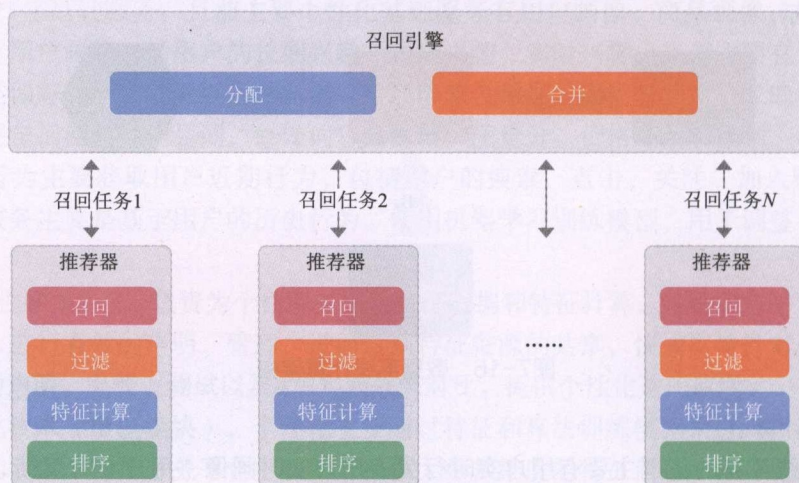


图7-18 推荐引擎技术架构

分配。根据推荐场景，按召回源进行任务拆分，关键是让分布式任务到达负载均衡。

推荐器。推荐引擎的核心执行组件，获取个性化推荐结果，推荐器的实现如图7-19所示。

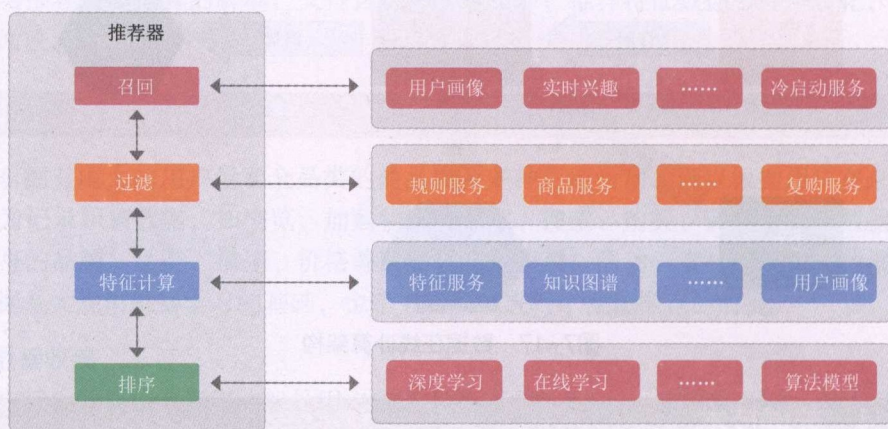


图7-19 推荐器架构

◆ **召回阶段。**获取候选集，一般从基于用户画像、用户偏好、地域等维度进行召回，如果是新用户的召回资源不够，会使用冷启动服务进行召回。

◆ **规则过滤阶段。**对人工规则、一品多商、子母码、邮差差价等进行过滤。

◆ **特征计算阶段。**结合用户实时行为、用户画像、知识图谱、特征服务，计算出召回

的候选集的特征向量。

◆ **排序阶段。**使用算法模型对召回候选集打分，根据召回源和候选集的分值，按一定的策略对候选集进行重新排序。

合并。归并多个推荐器返回的推荐结果，按业务规则进行合并，考虑一定的多样性。

举例来说，京东App首页“猜你喜欢”的实现过程如图7-20所示。首先根据用户画像信息和用户的近期行为及相关反馈信息，选择不同的召回方式，进行业务规则过滤；对满足要求的候选商品集，提取用户特征、商品特征、用户和商品的交叉特征；使用算法模型根据这些特征计算候选商品的得分；根据每个商品的得分对商品进行排序，同时会丰富推荐理由，考虑用户体验，会对最终排好序推荐结果进行微调整，如多样性展示。

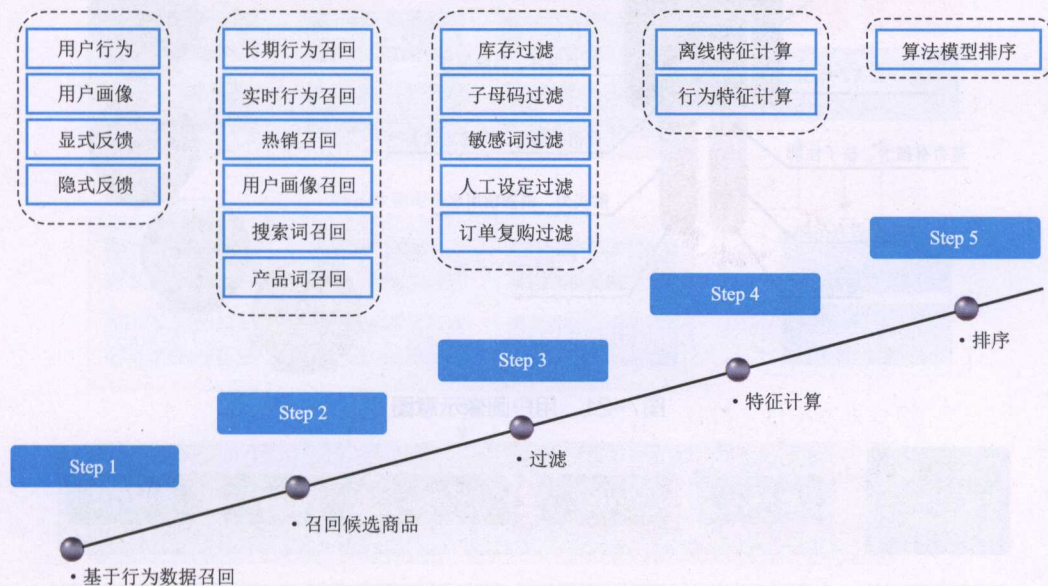


图7-20 猜你喜欢实现过程图

2. 用户画像

京东大数据有别于其他厂商的地方就是京东拥有最长的价值链和全流程的数据积累。京东数据的特征非常全面，数据链记录着每个用户的每一步操作：从登录到搜索、浏览、选择商品、页面停留时间、评论阅读、是否关注促销，以及加入购物车、下订单、付款、配送方式，最终是否有售后和返修，整个用户的购物行为完整数据都被记录下来。通过对这些用户行为及相关场景的分析，构建了京东用户画像，如图7-21所示。

其中不仅有用户的年龄、性别、购物习惯，更有根据其购物行为分析出的大量数据，例如是否已婚，是否有孩子，对促销是否敏感等。另外，实时用户画像可以秒级分析出用户的购买意图，以及实时兴趣偏好。京东推荐用户画像技术体系如图7-22所示。

用户画像在京东各终端的推荐产品中都有应用，618推出的智能卖场是用户画像的典型应用场景。智能卖场的产品包括发现好货、个性化楼层、秒杀、活动、优惠券、分类、标签等。以秒杀为例，推荐结果会根据当前用户的用户画像中的画像模型（性别、年龄、促



销敏感度、品类偏好、购买力)进行加权,让用户最感兴趣的商品排在前面。

用户画像也是场景推荐的核心基础。以东家小院为例,根据用户的历史行为汇聚出很多场景标签,按当前用户的画像模型,调整场景标签的排序。如用户选择“包治百病”标签,会按用户画像中的性别、年龄、品类、促销敏感度等画像模型进行推荐商品的重排序。

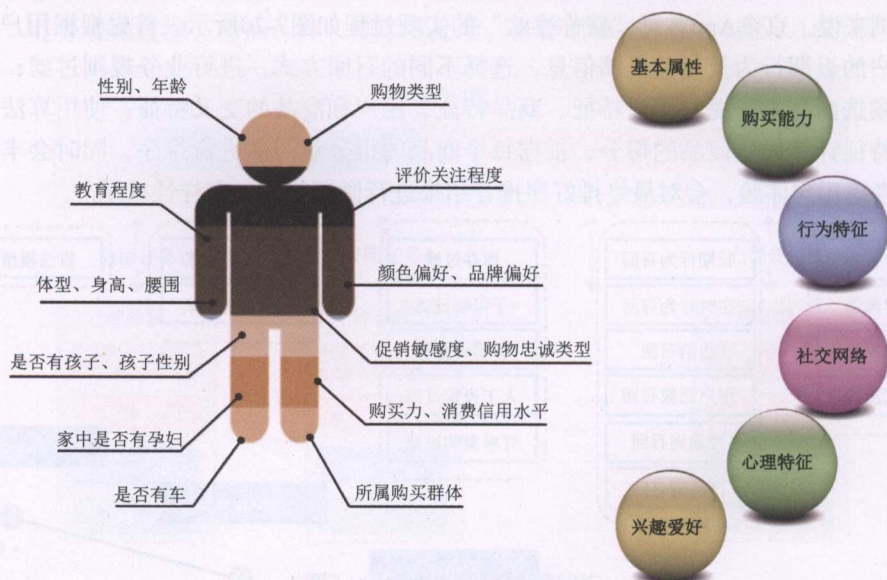


图7-21 用户画像示意图

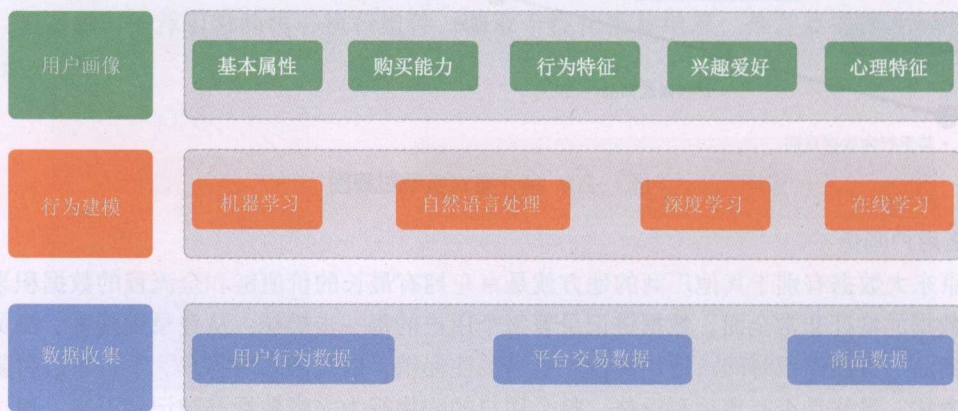


图7-22 京东推荐用户画像技术体系

3.特征服务平台

特征就是一种属性的描述,特征是个性化推荐的基础,常用的特征分为单边特征和双边特征。单边特征是指对象本身的属性描述,如商品的颜色;双边特征是指两个对象交互程度的描述,如某用户最近一小时浏览的品牌与候选集中品牌的匹配程度。从特征生成的场景来说,分为离线特征和实时特征。离线特征是通过算法模型提前生成,实时特征是通过实时计算的方式生成的。特征的质量直接影响推荐的效果、特征计算的性能,同时影响个性化推荐

的处理能力。另外，共享和复用特征可以提高算法的迭代速度并节约人力成本。

特征服务管理平台主要针对特征数据和特征计算，进行有效声明和管理，进而达到特征资源的共享和复用。特征服务平台能快速满足针对制定不同的特征进行有效的声明、上线、测试以及A/B实验效果对比的需求，做到特征的可维护、可说明、可验证。特征服务平台的主要功能如下：离线特征的定制化使用，在线特征的定制化使用，由定制化特征产生新的特征，部分特征、模型在线申明，不同特征效果快速A/B。特征服务平台架构如图7-23所示。

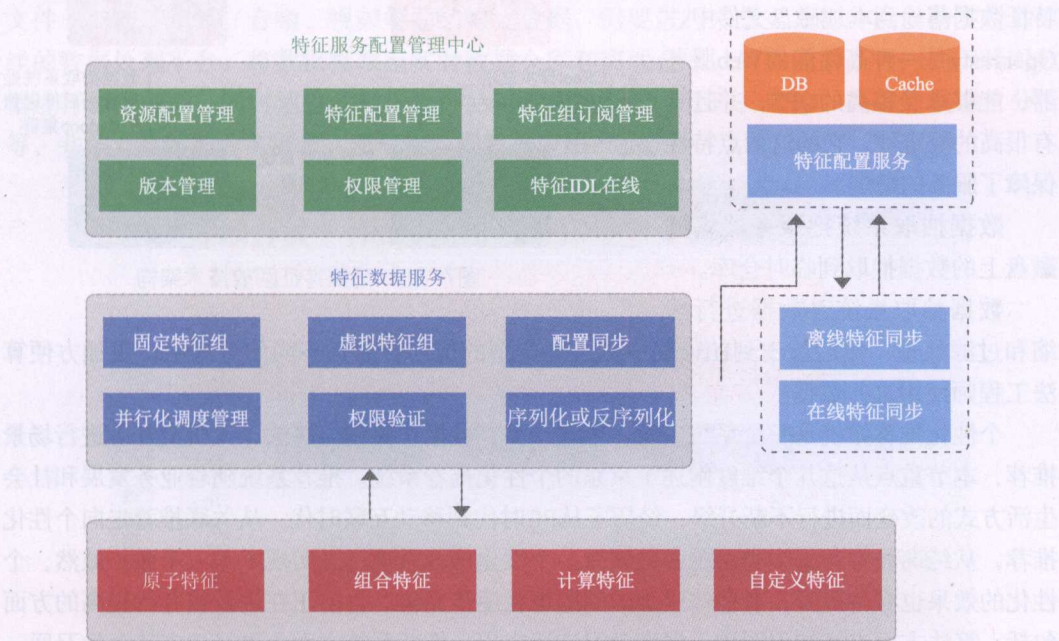


图7-23 特征服务平台架构

4.场景特征回放技术

推荐的一般处理逻辑是每次请求会召回一批商品，然后根据用户的行为数据和用户模型计算出每个商品的特征。算法模型会根据每个商品的特征计算出每个商品的得分，最后选出得分最高的几个商品推荐给用户。

线上计算特征这种行为是一次性的，不会被记录下来。因此在线下训练模型的时候，如果想利用上述的特征，就需要在线下机器上再次计算一遍这些特征。遗憾的是，线下计算出来的特征往往不能和线上特征完全相同，这就导致了模型训练的效果较差。场景特征回放示意图如图7-24所示，推荐业务调用推荐引擎，推荐引擎将场景特征通过特征回放服务记录下来，推

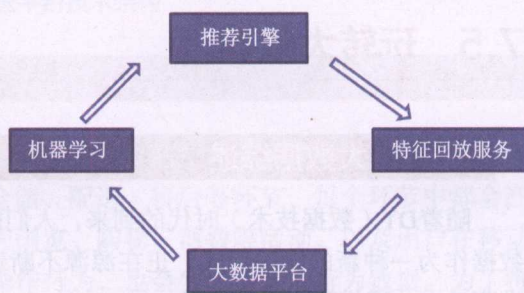


图7-24 场景特征回放示意图



送至大数据平台，机器学习根据场景特征数据重新训练算法模型，进而影响推荐引擎中的排序，形成一个场景闭环推荐，达到更准确的个性化推荐。

场景特征回放技术架构如图7-25所示，场景特征回放技术实现过程如下。

线上特征一般是一系列的数值，我们将这些特征按照一定的规则组装成一个字符串，然后将特征使用HTTP的POST方法异步发送到服务端。

服务端使用Openresty接收这些HTTP请求，并把HTTP请求中的特征数据落地到本地磁盘文件中。Openresty是一种高性能的Web服务器，能够承受很高的QPS，并且具有很高的稳定性，它的这两点特性保障了服务的稳定。

数据抽取系统把服务器集群磁盘上的数据抽取到临时仓库。

数据抽取系统对数据进行压缩和过滤处理，然后推送到Hive表中。不同类型的请求会放到不同的分区中，更加方便算法工程师使用这些数据。

个性化推荐系统是一个系统工程，依赖产品、数据、架构、算法、人机交互等进行场景推荐，本节重点从这几个维度阐述了京东的个性化推荐系统。推荐系统随着业务发展和生活方式的改变而进行不断升级，经历了从PC时代到移动互联时代，从关联推荐走向个性化推荐，从纯商品推荐到多类型推荐的转变。个性化推荐系统已经实现了千人千面。诚然，个性化的效果也有待提升，有些体验类的问题也在逐步完善。目前正在进行或有待提高的方面包括：算法方面丰富知识图谱、深度学习广泛应用；推荐系统方面会更好地支持海量召回、高维特征计算、在线学习，推荐更实时，更精准；产品方面已向“满屏皆智能推荐”方向迈进。最后，希望个性化推荐系统能让购物变得简单，变得更人性化、更丰富、更美好。

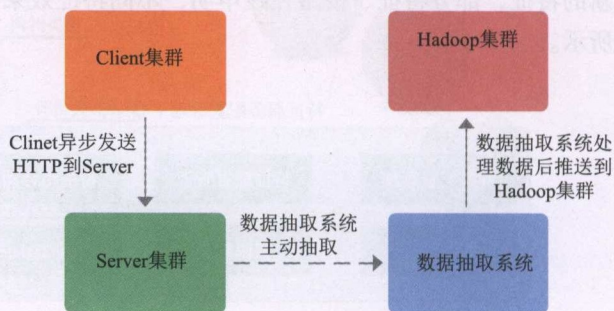


图7-25 场景特征回放技术架构



fisherman：时任推荐部门推荐系统负责人，负责推荐部门的架构设计及相关研发工作。

7.5 玩转大数据

7.5.1 大数据综述

随着DT（数据技术）时代的到来，人们能比以往更容易、更多地获取更丰富的数据。数据作为一种新的能源形式，正在源源不断地发挥其巨大的价值，帮助我们激发更多的技术驱动力，提供更优质的服务。

在京东，有着EB级规模的历史数据，每天有近PB级的数据增长，同时每天有百万级的

数据处理任务在执行。数据井喷式的增长给数据采集、数据处理、数据管理、数据应用、数据质量、数据运维带来了极大的考验。

京东的数据目前包含了电商、金融、广告、配送、智能硬件、运营、线下、线上等场景的数据，每个场景的数据背后都存在着众多复杂的业务逻辑。为了帮助业务人员降低获取数据的门槛，简化数据获取的流程，同时帮助分析人员方便快捷地进行数据统计分析，进而挖掘数据的潜在价值，京东搭建了一套完整的数据解决方案。如图7-26所示为大数据平台技术架构，分散在四处的线上系统数据（多为结构化的业务数据），或者是各种日志文件、文档、图片、音频、视频等非结构化数据，需要进行采集。我们分别借助实时和离线的数据处理平台，将数据抽取至实时数据仓库和离线仓库，然后借助平台内的工具对数据进行加工处理，同时辅以各种平台产品对数据进行统一管理、监控、处理、查询、分析等，并结合具体的业务需求，形成相应的数据应用产品。

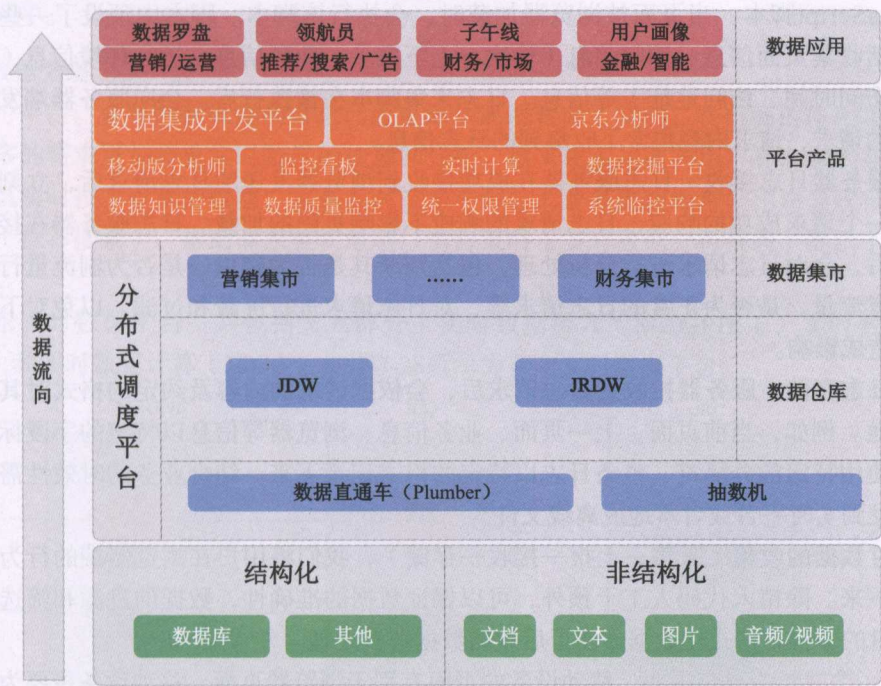


图7-26 大数据平台技术架构

7.5.2 技术平台

1. 数据采集

京东包含了电商所涉及的营销、交易、仓储、配送、售后等环节，每个环节中都会产生大量的业务数据，同时用户在网站上进行的浏览、购物、消费等活动，以及用户在移动设备上对应用的使用情况，包括各种系统的操作行为，也会生成海量的行为数据。为了将上述的结构化业务数据以及用户非结构化的用户行为日志进行采集，京东搭建了一套标准化采集方案，能够将业务分析所需的数据进行标准化采集，并将数据传输到大数据平台，



以便后续的加工处理及上层的数据应用。

目前京东的数据采集方案主要分为两大类：用户行为日志采集方案（点击流系统）和通用数据采集方案（数据直通车），下面将做详细介绍。

点击流系统。目前京东有着丰富的入口平台及展示形式，包括PC网页、H5页面、App应用、App内部的H5页面、智能设备、微信手Q以及微信生态下的新场景微信小程序。其中PC网页、H5页面、App内部的H5页面、微信手Q以及微信小程序由网页方式呈现，用户通过浏览器进行访问；而智能设备，例如手机、移动手环、智能家电等，则是以App应用的方式呈现，用户访问App即可获得相应的服务。以下是浏览器和App应用两种使用场景的日志采集方案。

◆ 浏览器端的日志采集

①日志采集。浏览器的日志采集方式，首先需要在统计页面日志的页面中预先植入一段JavaScript脚本，当页面被浏览器加载时，会执行该脚本。脚本中预设了一些采集需求，包括收集页面信息、访问信息（访次、上下文）、业务信息、运行环境信息（浏览器信息、访问时间、访问地址）等信息。日志采集脚本在被执行后，会向服务器端发送一条HTTPS的请求，请求内容包含了收集到的日志信息。

②服务器日志接收。日志服务器在成功接收到浏览器发送的日志请求后，立刻向浏览器发送一个请求成功的响应，日志请求的响应不影响页面的加载。日志服务器在接收到日志请求后，会对日志请求进行分析处理，包括判断其是否为爬虫、是否为刷流量行为、是否为恶意流量、是否为正常的日志请求等，对日志请求进行屏蔽和过滤，以免对下游解析和应用造成影响。

③日志存储。服务器接收到日志请求后，会依据请求的内容及约定的格式对其进行格式化落地。例如，当前页面、上一页面、业务信息、浏览器等信息以特定的字段标识，字段之间使用特定的分隔符，整条日志以特定的格式记录下来。结合业务的时效性需求，将日志分发到实时平台或者落地成离线文件。

经过数据的收集（采集—上报—接收—存储），我们将用户在浏览器端的行为日志实时记录下来。除植入代码人工干预外，可以保证数据的准确性，数据的过滤和筛选保证了异常流量的干扰，格式化数据方便了后续的数据解析处理。

◆ 移动设备的日志采集。移动设备的页面有别于浏览器页面，移动设备主要为原生页组成的App应用，原生页使用原生预研开发完成。例如Android系统使用Java语言，iOS系统使用Objective-C原生语言开发，原生页运行速度快，效率高。

①采集方式。移动设备上App应用的数据采集主要使用的是SDK工具，App应用在发版前将SDK工具集成进来，设定不同的事件行为场景，当用户触发相应的场景时，则会执行SDK相应的脚本，采集对应的行为日志。

②日志存储。用户的各种场景都会产生日志，为了减少用户的流量损耗，我们将日志先在客户端进行缓存，并对数据进行聚合，在适当时机对数据进行加密和压缩后上报至日志服务器，同时由于数据的聚合和压缩也可以减少对服务器的请求情况。

数据直通车。数据直通车为京东线上数据提供接入京东数据仓库的完整解决方案，为后

续的查询、分发、计算和分析提供数据基础。直通车提供丰富多样、简单易用的数据处理功能，可满足离线接入、实时计算、集成分发等多种需求，并进行全程状态监控。

如图7-27所示的数据直通车接入数据类型，根据抽取的数据量及抽取对线上的影响，会分为定时的离线接入和实时接入两种抽取方式。每种抽取方式支持不同的数据类型，每天在零点后可以获取前一天完整的数据，然后将一整天的数据进行集中加工处理，并将数据最终储存到目标表对应的分区中。



图7-27 数据直通车接入数据类型

2. 数据处理

实时平台。业务数据处理的需求已经逐渐从离线转向了实时，在电商的应用场景中，越来越多的需求更加倚重实时数据的处理和分析，越来越多的面向用户和商家的业务场景开始尝试实时技术带来的收益。京东实时技术平台协助业务更快地帮助用户发现自己想要的商品（推荐搜索），帮助商家更快地制订销售策略（实时数据分析报表）提供了强有力的支撑。

京东实时数据平台一共包括三大部分：实时数据接入（MAGPIE），实时数据传输（JDQ）和实时数据计算（JRC）。如图7-28所示为京东实时数据平台。

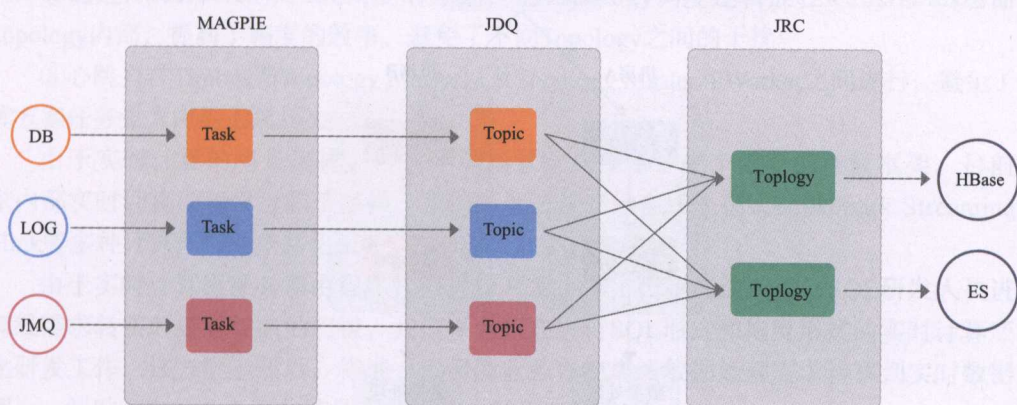


图7-28 京东实时数据平台

下面就实时数据处理分析在京东的技术流程进行阐述。

◆ **实时数据接入。**实时数据的源头是各个线上业务系统的各种类型数据源，在京东内部主要包括三个部门：

①线上业务系统数据库：MySQL、SQL Server、Oracle。目前京东内部线上系统基本都已经切换MySQL。实时数据接入系统Magpie完全支持上述三个关系型数据库的数据实时接入，原理为数据库的主从复制模式，通过伪装从库的方式，把关系型数据库的Binlog



日志实时抓取并解析发送到JDQ内。对于MySQL数据库，实时接入程序按照服务粒度抓取MySQL单服务上的所有Binlog，在程序内部进行Binlog的实时解析并过滤出所需要的库表，再发送到表粒度的Topic上，方便下游用户进行业务表粒度的实时处理。

②线上业务日志系统：统一流量（用户浏览点击日志），统一日志（各业务系统服务日志）。业务日志由线上系统先发送到JDQ的写集群，再由Magpie任务实时同步到JDQ的读集群。通过这种方式做到了日志数据的读写分离，极大地提高了系统稳定性和服务能力。

③线上消息系统：JMQ。JMQ是京东内部线上系统的消息中间件服务，很多业务数据在落数据库之前都会经过JMQ系统在不同业务系统之间进行传递。Magpie同样可以把JMQ内的线上系统消息实时地同步到JDQ内，再面向数据处理用户进行消费，极大地提高了数据处理系统的服务能力。

京东内部的所有系统的实时数据都会经过Magpie系统进行接入和转发到JDQ系统，统一由JDQ对数据处理的业务需求提供消息服务。该方案帮助业务用户和技术层面屏蔽了接入的复杂度问题，并把服务稳定性和能力提高到了大数据实时处理的要求。

◆ **实时数据总线**。实时数据在由Magpie进行统一接入处理后，需要一个面向业务研发用户的消息消费服务。我们基于Kafka的JDQ服务就是满足这个需求的产品，如图7-29所示。在原生Kafka的基础之上，我们封装了权限、限速、监控报警等一系列服务。针对重要业务进行了双机房读写分离的部署方案，大大提高了消息服务的可靠性和服务能力。618当天日生产291TB，8000亿行数据，日消费1000TB。各个系统越来越重视通过日志进行数据分析，每次618的业务日志量均以150%的速度增长。

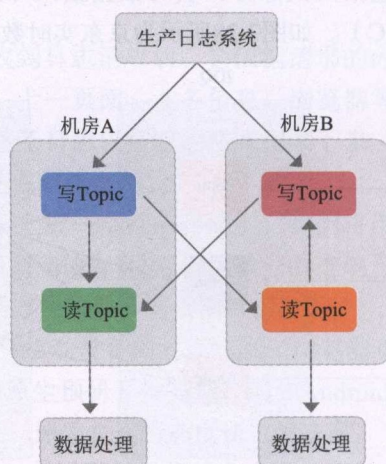


图7-29 实时数据总线

生产日志系统向最近机房内的JDQ系统的写Topic发送业务日志消息，如遇机房故障，自动切换到可用机房的服务。

JDQ系统通过实时同步不同写集群数据到每个机房的读集群，实现每个机房都有一份完整的业务日志数据可供业务研发消费。

业务研发就近机房选择读集群进行消费，同时通过JDQ可以实现不同用户的消费限

速，最大限度地保证集群服务的稳定可靠。

JDQ实时数据总线服务作为实时数据的中转缓存服务，屏蔽了业务研发对不同数据源的接入难度，同时通过一系列的数据格式使用方式的标准化，打通了实时数据从接入到业务处理的传输环节，实现了京东内部实时数据通道的目标。

◆ **实时数据计算。**实时数据要想体现业务价值，最终还需要业务研发方进行计算和分析。京东内部主流的实时计算平台是JRC计算平台，该平台脱胎于早期的Storm版本，由平台研发进行了深度的改造和产品化，实现了业务研发用户完全的Web产品任务管理和监控的需求，同时整合了JDQ数据来源，实现了用户在数据计算平台的无缝对接实时数据。本次618达到1.1万亿次日处理次数。

2017年618，JRC基于容器的新架构已经开始支撑部分线上业务，未来容器化的JRC方案会进一步提高Storm平台的稳定性和资源利用率。JRC架构图如图7-30所示。

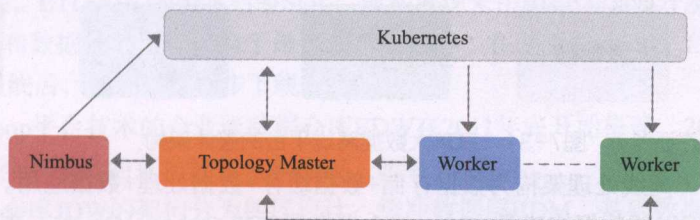


图7-30 JRC架构图

该方案的特点如下：

- ①通过Kubernetes实现Topology执行节点的容器化，资源随用随申请，提高资源利用率。
- ②通过Kubernetes和二级调度的方案，把Topology调度逻辑放在Kubernetes层面和Topology内部，提高了调度的效率，避免了不同Topology之间的干扰。
- ③心跳只在Timbus和Topology Master以及Topology Master和Worker之间进行，避免了传统方案任务量大时的心跳压力。

由于实时计算的场景多样，针对不同场景业内提出了多个流行的计算框架。目前京东内部实时计算的场景也趋于多样，我们平台已经开始在线上正式提供Spark Streaming和Flink等多种计算框架的产品化服务。

由于实时计算程序必须由程序代码进行开发，对于传统离线业务，SQL研发人员进行离线需求转实时还有较高的门槛，我们平台正在进行SQL形式和拖曳形式的实时计算产品化研发工作。该方案上线后，将进一步帮助业务方把离线数据处理需求转移到实时数据处理上，帮助京东的业务更快速地服务于广大的用户和商家。

目前京东实时数据解决方案整套流程已经接入了线上的上千张业务表数据流和数百个业务日志数据流，覆盖京东内部所有核心业务系统和大部分实时处理业务，主要面向京东内部各个业务部门的个性化推荐、秒杀、实时运营、商家报表等。未来，离线数据处理需求会越来越多地迁移到实时数据处理上。

离线平台。京东大数据离线平台的整体架构如图7-31所示。

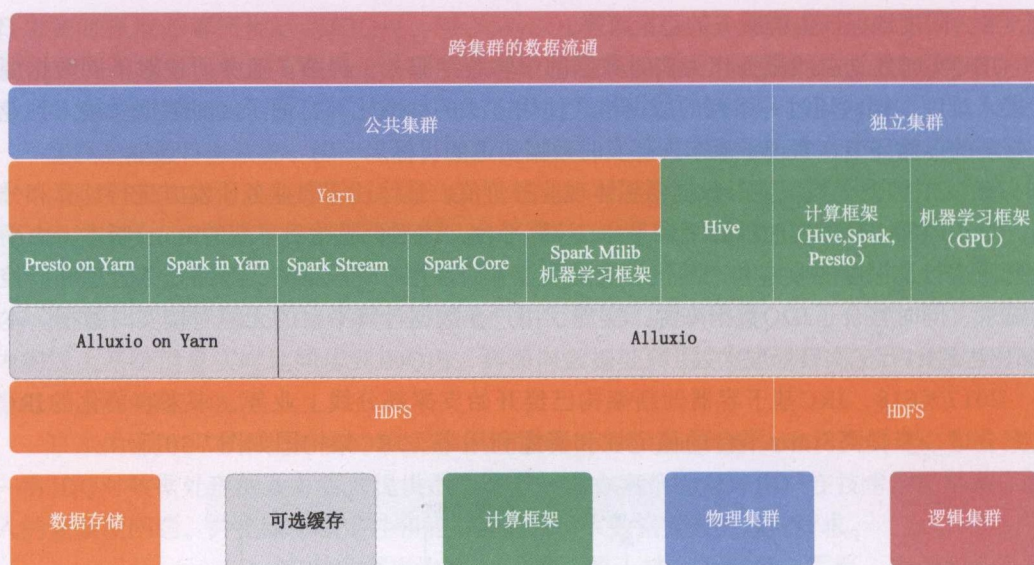


图7-31 京东大数据离线平台的整体架构

◆ **平台详解。**离线处理架构为数据存储+数据缓存+数据处理+数据应用。

①**数据存储。**以前数据仓库是LZO，线上业务是SQL Server、Oracle。现在数据仓库是ORC，线上业务是MySQL、HBase。

②**数据缓存。**Alluxio是一个基于内存的分布式文件系统，它是架构在底层分布式文件系统和上层分布式计算框架之间的一个中间件，主要职责是以文件形式在内存或其他存储设施中提供数据的存取服务。

③**数据处理。**混合型引擎，按需按量分配，以及根据不同业务场景，选择不同处理方式，统一由Yarn做资源管理。

④**数据应用。**服务京东消费数据的几乎所有场景，如数据挖掘、分析报告、常规报表、即席查询等。

◆ **具体介绍。**在京东大数据平台中有多个物理集群、十几个集群应用软件、十几个大数据产品、30多个数据集市、6000多个平台用户，日运行job数量超过40万，日计算数据量超过15PB。在如此庞大的业务场景、海量数据计算、复杂数据处理流程的场景下，一个高效实用的大数据离线平台显得尤为重要。

为此，我们对大数据平台建设以来支持的各类业务服务，大数据平台自身的升级与运维技术工作进行了梳理分析，对大数据平台从前端服务到后台技术进行了整体服务框架设计。完成了从多出口的臃肿服务到统一服务管理、自助化服务管理、自动化服务实现的有机“瘦身运动”，大数据平台服务时效得到了几倍乃至几十倍的提升。

大数据平台已经实现了海量数据的实时与离线计算，同时也达到高并发、高容错、高扩展、低成本的集团发展需要。同时，在保证现有大数据平台稳定的基础上，通过与京东集市三十多个业务集市的深入接触沟通，在业务发展基础上，结合最新、最适合的前沿技术，不断提高大数据平台的业务实现范围、大数据平台技术创新（如异构集群、多引擎支

持、即席查询、多维分析、登月平台等）、大数据平台更好的运营管控机制（如大数据平台运营规范、数据仓库与集市建设规范、运营值班方案、流程中心等），不断满足京东业务高速发展对未来大数据平台的技术需要，实现京东战略价值目标。

7.5.3 数据仓库

1. 衍生之路

京东最早的数据仓库平台基于微软的产品工具平台SSIS（Microsoft SQL Server Integration Services），数据的ETL和展现均在平台内完成。由于缺乏模型层面的架构设计，随着后期数据量的增加，性能日趋下降。

2011年6月，基于Oracle数据库的全新数据仓库DW上线，模型设计分为ODS、DWD、DW、App四层，ETL采用Tcl脚本封装SQL，前端展现采用BIEE和Java开发的灵活报表。随着业务需求的和数据量的增长，由于昂贵的扩展成本，在基于Hadoop平台技术的企业级数据仓库EDW上线后，于2012年逐步下线。

基于Hadoop平台技术的企业级数据仓库EDW在2011年底开始搭建，2012年上线，后在2013年升级为全新设计的京东数据仓库JDW，如图7-32所示。

京东数据仓库JDW的架构分为四层设计：缓冲数据层BDM、基础数据层FDM、通用数据层GDM、聚合数据层ADM。除基础数据层外，还包括维度层DIM以及各个数据集市的应数据模型App，每个数据层的用途见表7-3。

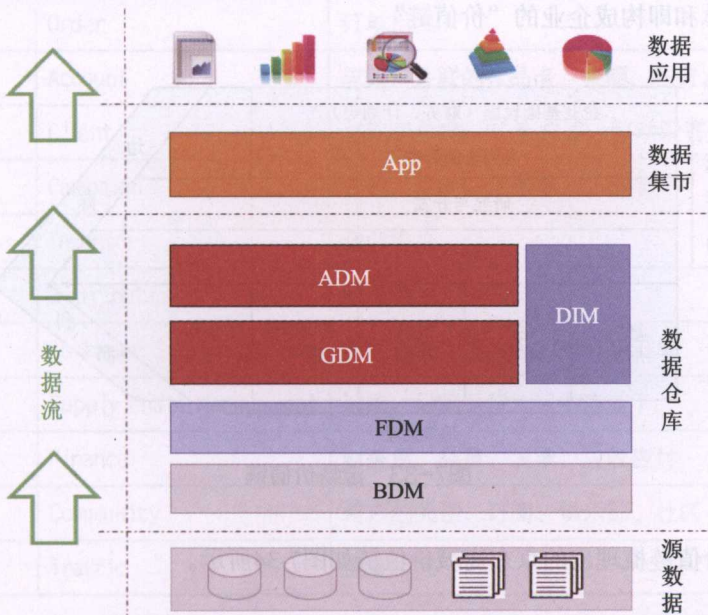


图7-32 京东数据仓库JDW架构



表7-3 京东数据仓库JDW的架构分层及用途

模型分层	简称	数据层次用途
缓冲数据层	BDM	源业务系统数据的快照，保存细节数据，按天保存
基础数据层	FDM	按业务概念组织细节数据，流水或拉链设计
通用数据层	GDM	根据京东核心业务价值链，按照维度建模的星型模型或雪花模型设计方式建设的最细业务粒度汇总层。在本层需要进行指标与维度的标准化，保证指标数据的唯一性
聚合数据层	ADM	根据不同的业务需求，采用星型或雪花型模型设计方法构建的数据集市
维度层	DIM	维度是对具体分析对象的分析角度，维度要具备丰富的属性，历史信息的可追溯性，对通用的维表要保持一致性

2.模型体系

企业的价值创造是通过一系列活动构成的。这些互不相同但有互相关联的生产经营活动，构成了一个创造价值的动态过程，即波特价值链，如图7-33所示。

◆ 基本活动包括进料后勤、生产作业、发货后勤、市场和销售、售后服务等。

◆ 辅助活动包括采购、研究与开发、人力资源管理和企业基础设施（财务、计划等）等。

企业要生存和发展，必须创造价值。如果把“企业”这个“黑匣子”打开，我们可以把企业创造价值的过程分解为一系列互不相同但又相互关联的经济活动，或者称之为“增值活动”，其总和即构成企业的“价值链”。

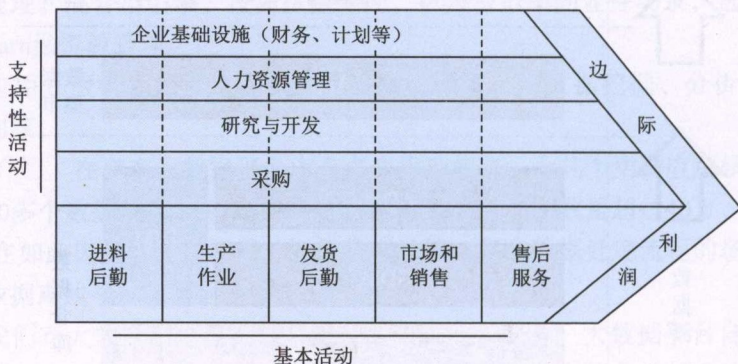


图7-33 波特价值链

根据波特价值链梳理出的京东商城价值链如图7-34所示。

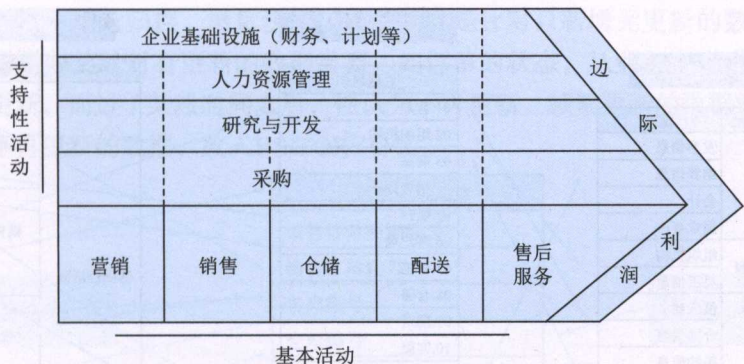


图7-34 京东商城价值链

基于京东企业价值链设计，我们将京东数据仓库JDW的模型体系划分成14个主题，见表7-4。

表7-4 京东数据仓库主题模型

主题划分	主题英文名	覆盖内容（包含但不限于）
01. 客户	Customer	供应商、POP 商家、团购商家、合作伙伴、用户
02. 组织机构	Organization	员工、部门
03. 商品	ITEM	商品相关
04. 销售订单	Order	订单相关
05. 账户	Account	与账户关联的礼品卡、余额、积分、优惠券等
06. 客户端	Client	移动客户端、PC 客户端、移动设备
07. 营销	Campaign	促销、活动、优惠券、礼品卡
08. 仓储	Inventory	仓储相关
09. 配送	Distribution	配送相关
10. 客服	Customer ServiceCenter	售后、备件库、呼叫中心、工单
11. 供应链	Supply Chain Management	采购、采购退货（退供应商）
12. 财务	Finance	财务单、结算、发票、应收应付
13. 社区	Community	用户的关注、订阅、俱乐部、社区
14. 流量	Traffic	流量相关

京东商城价值链与主题模型的映射关系如图7-35所示。

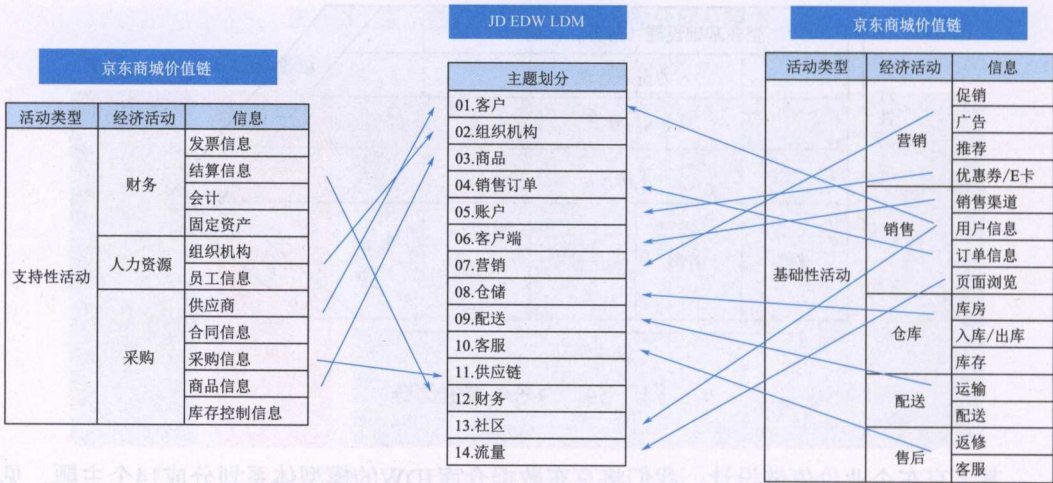


图7-35 京东商城价值链与主题模型的映射关系

3.模型设计

◆ **模型设计理论。**JDW的通用模型设计以采用Kimball的维度建模。维度模型设计期间主要涉及4个主要的决策：业务过程、粒度、维度、事实。业务过程是组织完成的操作型活动，如获得订单。确定粒度是维度模型设计的重要步骤，粒度用于确定某一事实表中的数据细节程度。维度提供围绕某一业务过程事件所涉及的“谁、什么、何处、何时、为什么、如何”等背景。事实涉及来自业务过程事件的度量。星型模型是部署在关系型数据库之上的多维结构，主要包含事实表，以及通过主键/外键关系与之关联的维度表。

企业数据仓库总线架构提供了一种分解企业数据仓库规划任务的合理方法。企业数据仓库总线矩阵是总线矩阵的文档化。如图7-36所示为京东订单明细总线矩阵示意。

业务过程	公共维度					
	订单日期	仓库	配送区域	客户	促销	商品
提交订单	×		×	×	×	×
支付					×	×
打包						×
分拣						×
出库		×				×
配送			×	×		×
收货				×		×

图7-36 京东订单明细总线矩阵示意

依据订单明细矩阵进行的订单星型模型示意图，如图7-37所示。

◆ **模型的策略。**对于DIM层的维表来说，一般设计不采用分区模式。对于聚合数据层ADM，一般是采用按日（或其他周期）增量设计，分区方式采用按日即可。对于通用数据层GDM来说，分区有两种类型，即全量设计和增量设计。对于主数据类型模型，如用户和商品，采用全量设计。对于交易数据类型，如订单，采用增量设计。增量设计细分为3种，

分别是增量流水、增量归档、增量快照。增量流水是针对只新增无更新的数据类型，采用按日分区。增量归档针对有更新的数据类型，如订单的状态，从业务上来讲，这种数据在近期会频繁更新，而过了某段时间之后，便认为不再更新。频繁更新的数据放入ACTIVE分区，而认为不再更新的数据，放入HSITORY分区。

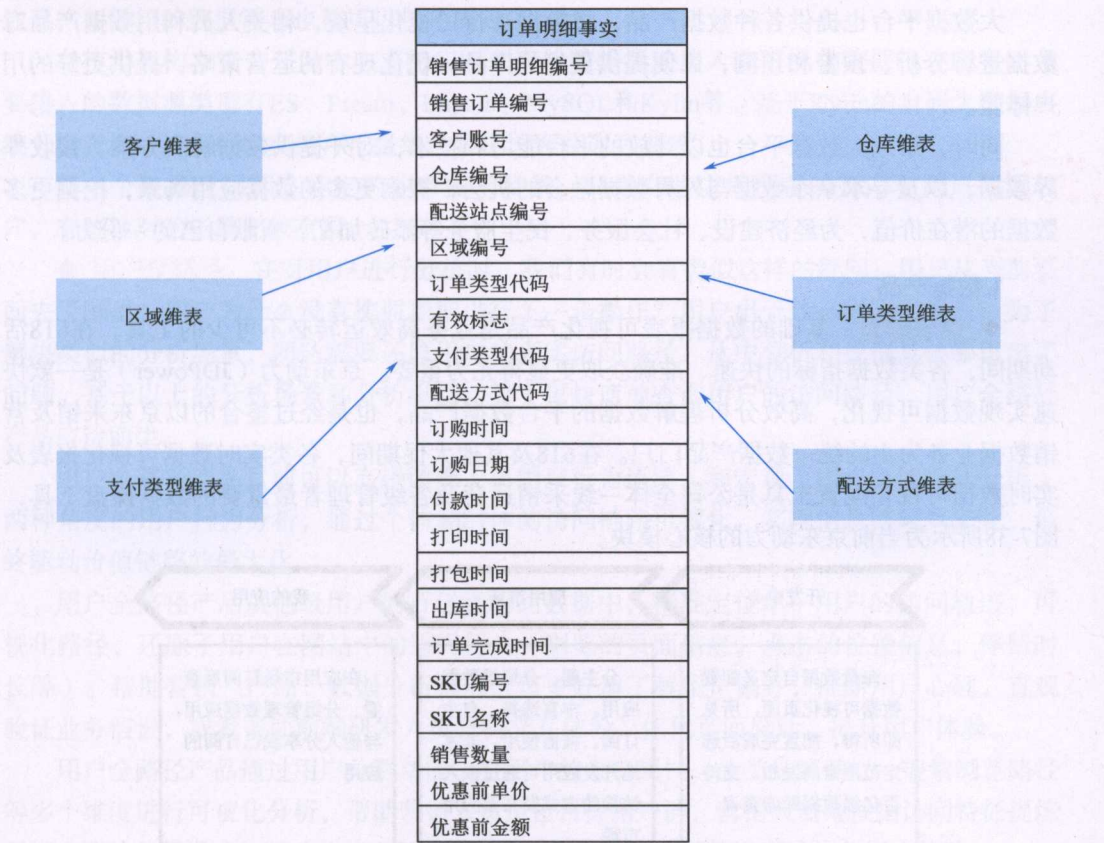


图7-37 订单星型模型示意图

增量归档方式有一个缺点就是某些数据并没有一个真正的关闭状态，即使过了很久甚至归档了仍然会被更新，这样就会造成数据重复。还有一个不能满足需求的地方是数据是动态变化的，在某些情况下还原历史状态会存在一定困难，如无效，会造成前后统计的数据存在一定的差异，增量快照的设计方式可以避免这两点，既不会重复，也可以方便查询历史状态数据。增量快照的设计方式是把当日的增量数据保存到当日分区，历史分区数据不再更新，通过增量数据，既可以统计新增的时间，也可以统计每天各个环节的数据。

7.5.4 数据应用

京东大数据平台通过数据采集工具将数据收集到数据仓库中，在数据仓库中对数据进行规范管理，再通过实时平台与离线平台将数据进行加工处理，使烦杂的数据转变成业务所需要的结构化基础数据。



京东大数据平台在集合了所有的业务数据后，将数据应用到各种业务实践中，如将数据提供给推荐，用于更精准的个性化服务，优化购物体验；提供给搜索，用于更优质的搜索服务；提供给广告，用于更精准的广告投放；提供金融、保险、信用等金融平台，用于获取更好的金融服务。

大数据平台也提供各种数据产品，将数据进行可视化呈现，相关人员利用数据产品对数据进行分析、预警和预测，以便提供更好的产品，优化现有的运营策略，提供更好的用户体验。

同时，京东大数据平台也以开放的平台能力和技术，对外提供数据解决方案，接收外界数据，以及寻求京东数据与外界数据融合的机会，探索更多的数据应用场景，挖掘更多数据的潜在价值，为经济建设、社会服务、民生服务等添砖加瓦，贡献自己的一份力。

1. 数据产品

◆ **京东动力**。基础的数据报表可视化产品是业务高效运转必不可少的工具，在618活动期间，各类数据指标的快速、准确获取更显得尤为重要。京东动力（JDPower）是一款快速实现数据可视化，高效分析理解数据的平台数据产品，也是经过整合的以京东采销及营销数据业务为主的统一数据产品门户。在618及其他大促期间，各类实时数据可视化报表及实时数据可视化配置工具是公司全体一线采销运营及各级管理者最重要的数据操盘工具，图7-38所示为当前京东动力的核心模块。



图7-38 京东动力核心模块

京东动力包含三大核心模块：开发中心、应用市场和我的应用。看到AppStore再联想自己安装手机应用的过程，或许就已经理解了这个产品的总体架构了。是的，各类数据可视化报表最终都会以应用的方式存在于应用市场和我的应用当中，而这些应用的来源主要是开发中心。

产品设计之初，主要解决两大业务痛点，首先是整合众多的数据产品，统一访问入口，统一数据分析逻辑，统一数据指标体系。另一个是快速实现各事业部的数据可视化需求。由于京东的业务扩展速度非常之快，且原有业务也都会不断地进行策略性调整，因此对于数据报表可视化的产品需求场景也是非常多样化的。2016年年初进行数据团队整合时已经发展出13套不同的数据产品，另外还有一些功能系统同时也在提供数据产品服务。在

这种局面下，对于需要各种数据的一线业务人员来讲可想而知是多么痛苦的，他们每天要花相当多的时间在不同的系统之间切换，查业绩、看库存、看用户数据、商品数据、流量数据等。前期产品发布时就采用嵌入链接等方式，先将各类数据产品发布到应用市场，用户订阅后通过我的应用统一访问。而具备自助化配置数据可视化页面功能的开发中心上线之后，各部门的数据需求也都可以快速满足了。

在技术架构方面，开发中心支持实时和离线数据源的接入和可视化数据报表配置。主要接入的数据源类型有ES、Presto、Impala、MySQL和Kylin等。基于Kylin的自研大数据多维分析查询引擎更可以支持百亿量级的数据即席获取和展现。

每一个数据应用都有特定的业务需求场景，只是有些应用覆盖的部门和用户范围较广，有些针对的场景比较有限。

◆ **用户全路径。**在对用户进行分析时，我们有时会有类似这样的疑问：用户从当前页面去了哪里？用户为什么没有按照预期进行下一步操作？用户出于什么原因而丢失？为了解决类似的分析场景，团队需要还原用户的真实访问路径，从中分析和挖掘具体哪里出了问题。基于以上的分析场景和分析思路，为了更快速地查询用户的访问路径，用户全路径产品应运而生。

用户全路径产品目前可以支持微观（单用户行为还原）和宏观（用户群体特征分析）两种角度的用户行为分析，通过个体和群体的访问链路可视化，实现信息链路具象化，最终驱动价值链路的最大化。

用户全路径产品从亿级用户的百亿级访问数据中，精准定位单一用户的访问轨迹，可视化路径，还原了用户在网站中的访问行为（浏览的页面信息，点击的位置信息，停留时长等）。帮助营销、产品、数据分析师等人员更好地了解用户偏好，推断用户心理，直观验证业务假设，或者帮助前端研发人员分析交互体验，优化页面设计，提升用户体验。

用户全路径产品通过用户分群功能，将用户的人口属性、品类品牌偏好、搜索浏览路径等多个维度进行可视化分析，帮助营销人员定位目标用户群，营销人员则根据访问特征提炼目标人群的关键诉求，驱动业务进行策略优化，不断提高营销精准度，优化用户体验。

例如，当有品牌商准备入驻京东，希望了解京东用户的一些浏览、搜索、关注行为、购买人群画像等特征，分析师在得到具体的分析结果后，可以在用户全路径上进行人群细查功能进行人群验证，同时也可以借助Query分析功能对搜索词等数据验证，确定分析结果的可靠性后，向品牌商提供最终的分析报告。

◆ **精准营销平台MKT。**精准营销平台MKT，是一款分析密集型、营销智能型的数据产品，集成了用户洞察分析、人群创建与管理、营销任务创建与管理、效果分析四部分内容。贯穿了营销的全流程，帮助营销人员告别了粗放型广告投放阶段、凭经验手工提取营销用户的烦琐阶段，最大限度地提升了营销效率和营销精度。

其中，用户洞察分析是营销的前置分析阶段，通过该模块，展现新老用户群的结构化特征、用户画像属性、品类品牌关联购买行为，帮助业务人员全面把握用户动态，制订营销策略和方案。

人群的创建包括标签筛选、场景营销和人群上传等多种类型，可以通过丰富的用户画



像标签池进行条件组合筛选，也可以通过机器学习算法，根据品类或品牌的营销场景智能识别目标用户，方便对人群进行营销推广，提升转化。

通过任务管理，对于创建的人群，可以直接对接营销渠道，包括短信、EDM、优惠券以及线上业务系统等，完成个性化营销。同时，跟进用户的效果统计，对比触达前后的用户浏览和下单表现。对于多次营销任务积累下来的效果数据，可以变成样本数据，反馈到数据系统，迭代优化模型特征，提高人群的智能筛选准确率。

例如，生鲜事业部同事需要进行新用户拉新，可以在产品中选择拉新场景，然后开始筛选当前场景下的用户特征，选择近期在生鲜品类上的行为特征，如有浏览、搜索过、关注过生鲜品类；选择用户画像的信息，例如某个地域的、收货地址、性别；选择用户的价值信息，如过滤风控用户；选择高价值人群，用户会员级别等，选择用户曾经被触达的情况，如邮件、短信等。在进行筛选时，可以实时获取满足当前条件的人群数量，业务人员结合当前的人群数量可以动态调整当前筛选的条件，在满足条件的人群中，可以再设定具体的用户数量，然后对部分用户进行后续的营销触达。

2. 数据分析与挖掘

京东大数据业务的数据分析与挖掘是基于京东全产业链的业务背景展开的，结合对业务全方面深入支持而来。从业务流程上来讲，它涵盖商品采购、广告推广、精准营销、仓储、配送、客服、售后、金融等八大关键节点。结合业务场景的深入洞察，极大地提升了整个业务链条的生产力。对于京东平台及相关业务方的业务提升起到了举足轻重的作用。

◆ **采购环节。**采购作为B2B电商的重要生产环节，精准的采购计划对于后续整个平台的成本、销售、毛利等核心财务指标的影响非常大。因此，基于动态的销售、配送以及售后等情况，结合采购商情况制定合理的采购计划，是整个供应链环节的重中之重。基于各大仓库区域的出库量预测、售后及备件库预测等，可以针对性地对采购商制订精准的采购计划。另外，通过建立宏观的采购指数，可以从整体、区域乃至行业上把握采购趋势，提供决策建议。

采购量会受到生产（销售）订单、出库订单、库存量等多方面的影响，其中出库订单在其中起到了承上启下的核心作用。对于全国各大区域每个仓库细化的品类或商品粒度的预测能够为采购计划提供重要信息。例如，团队预测华北区某仓库净化器出库量大幅度增长，可能引起该商品库存缺货，结合该商品采购周期过长等原因，为保证客户购买需求，可以临时安排临近仓库临时紧急内配、下调该商品营销投放、加大类似品营销活动或者紧急安排商品采购入库等措施，在均衡各项成本基础上，达成最优决策。

◆ **广告推广。**基于站内用户的行为特征和用户画像，对不同行业的品牌商特定的目标群体进行精准刻画和定位，并且与腾讯、今日头条等数据打通，对品牌商提供定制化的精准用户广告服务，并针对品牌商在定位潜在用户、提高用户转化、细分竞品人群、扩大市场份额等不同类型和场景需求方面提供个性化的服务支持。

团队基于京东丰富的品类信息，针对不同的品类特点，会为广告主量身定制专属的品类目标人群，从潜在购买、高价值客户、高兴趣群体、高关联品类等多种维度去洞察广告主需求。根据广告主基于自身品牌所属阶段，针对性地对其浏览、点击、购买、ROI等不

同的目标提供精准的用户群。例如，英特尔会根据其商品定位希望找到一些高端的设计师人群，团队会根据设计师的职业特点定位目标人群。第一步，找到一些确定性的少量种子人群；第二步，从他们对于绘画笔、高分辨显示器、高清显卡、设计师相关书籍，甚至在家居、个护、服饰等方面风格的差异来刻画人群特征；第三步，利用人群特征从而不断扩充设计师专属商品；第四步，重复第二步、第三步，循环迭代，在控制误差传播的基础上不断扩大精准的目标人群。在这个过程中，还可以定位出设计师人群感兴趣的其他相关品类商品、购买兴趣甚至地域等特征，这些都为英特尔在做品类联合促销、识别竞品、分地域推广等广告营销活动提供了非常实用的参考，对广告投放的点击转化率和ROI起到了很好的作用。

◆ **精准营销。**发挥对用户洞察方面的积累，针对用户在人口属性、购买潜力、购买意愿、品类和品牌偏好、购买力强弱、忠诚度、活跃度，以及营销渠道偏好、促销敏感度、生命周期等方面的行为偏好和表现，结合精准营销平台输出个性化精准营销方案，提高整个精准营销的效果，减少恶意骚扰，最大限度地提高用户体验。

◆ **仓储。**结合商品热销情况以及商品自身属性，如大小、易碎、高值等因素，综合考虑商品动销与现货情况，综合考虑商品损耗、缺货损失等因素，合理规划全国各大区域仓储布局、仓内摆放等情况。

如图7-39所示，针对仓储中对实现采购自动化、自动补货、订单均衡、共享库存等技术手段，提高整个仓储环节效率，从而提高京东对供应链的管控能力，提高人效，提升客户体验。

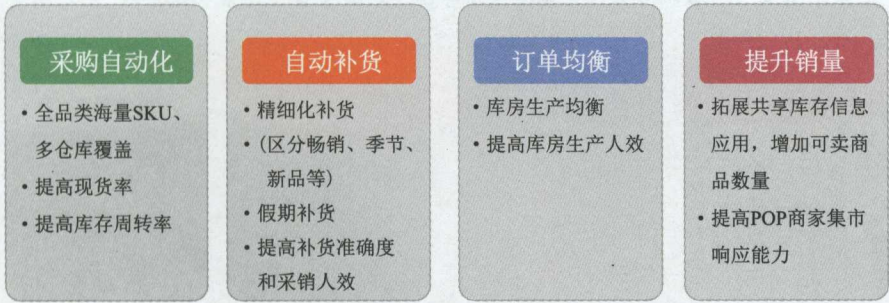


图7-39 仓储技术环节

就库房自动补货系统来说，根据商品历史销售记录，可以计算出每个商品的补货周期和最优补货量，可以每天在规定的时间内主动对某些SKU进行补货，实现库房商品从储物区向拣货区的最优转移，减少紧急补货次数，拣货区域商品周转率高。

◆ **配送、客服、售后。**在配送网格化、客服分级、售后先行赔付等方面进行客户分层和优化，提高配送时效、有效客服资源的快速响应、高信誉客户的先行赔付等数据挖掘洞察应用，有效提高了生产效率，提高客户体验。

例如，在配送环节，可以利用京东配送站的配送订单分布和配送路径来优化配送区域。以京东某一配送站为中心，用线连成一个区域，该区域覆盖了该配送站的所有订单，曲线圈成的面积为配送站覆盖的面积。配送员从配送站出发，到他配送的最后一单完成，



都会用一条折线连接起来，代表了配送员实际行走的路径，依据这些路径，我们可以优化配送区域。

如图7-40所示为某一配送站出发的配送路径示意图，每个点为用户收货地址，按配送先后顺序连线。每个城市根据这些配送站和其覆盖的区域可以对全市进行网格化划分，优化配送站选址和分布。

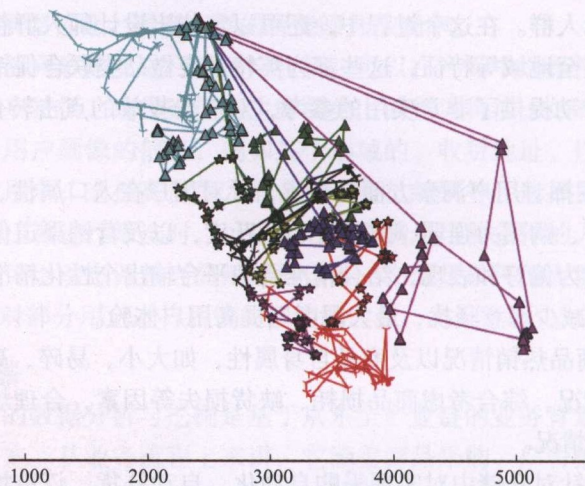


图7-40 配送路径示意图



看门老大爷：数据专家，负责点击流系统、实时平台、场景营销系统。

第8章

项目管理

运筹帷幄，万众一心

同舟共济扬帆起。要想打赢京东618技术攻坚战，强有力的项目组织协调必不可少，它让京东所有技术研发体系拧成一股绳，为同一个目标奋勇前进。





8.1 618项目管理挑战

2013年，京东在业界已经崭露头角，618之前，各个系统的负责人聚在一起，讨论如何备战。大家的一致看法是，618当天系统的流量可能非常大，零点时个别系统的峰值流量可能达到平时峰值的几十倍甚至上百倍。此时，京东系统数已经增长到几千个，涉及研发体系一级部门数近20个，部门之间的互相配合与统一组织迫在眉睫。于是，京东成立了“618技术备战项目组”，任命多名专职项目经理，以项目形式统一协调所有研发系统的备战工作。

8.1.1 项目管理的难度

京东拥有全球电商行业最全的业务系统，其电商系统的复杂程度，也许只有亚马逊可以与之相比。面向用户的不仅有PC和App两大渠道，还有微信手Q的一级入口。618来临之际，流量从各个渠道涌来，任何一个环节出现问题，都会被无限放大。

1. 项目组面临的挑战

备战618，项目组均面临着“三高”和“三多”的挑战：

◆ **重要性高**。618期间，用户的流量暴增，系统压力大，技术保障工作不容有失。对京东研发体系来说，就像一年一度的大考。

◆ **不确定性因素多**。历次618备战所处的环境及遇到的问题各不相同，所处的内外部环境也各不相同。例如2017年的备战，虽然技术团队已经提前制订启动了服务器采购计划，但恰逢全球SSD硬盘缺货，导致大量服务器无法按时到货。

◆ **头绪多**。项目组既有重大事项的推进，又有大量琐事需要安排。以安排集中值班座次安排为例，在会议室改造、桌椅采购并摆放完成之后，项目组要确定各团队的座次安排，需要综合考虑会议室布局、会议室容量、各团队值班人数、系统上下游依赖关系、系统重要程度等因素，尽量满足每个团队的需求，需要综合评估并适当取舍。

◆ **复杂度高**。京东业务极其复杂，备战工作涉及几千个系统、几万个应用的梳理、改造，以及应急预案的更新。

◆ **风险高**。平时运转良好的系统，在大流量的冲击下，也可能变得极其脆弱。例如有一次618，一个频道页平时的流量很低，从来没有出现过问题，但由于618当天促销火爆，流量太大导致该频道页宕机，并且由于事前没有准备该问题的应急预案，导致修复时间很长，造成了非常消极的影响。

◆ **干系人多**。直接参与项目的研发人员多达2000人，间接或部分参与该项目的人员也同样有2000多人。整个项目横跨京东商城、京东金融、京东物流三大子集团，以及商城研发体系、CMO研发、CTO研发等三大体系。

2. 项目经理面临的挑战

从项目管理的十大知识领域来说，每一个知识领域对项目经理都是极大的挑战。

◆ **范围**。各部门既有共性的备战需求，也有个性化的备战需求，需要做好范围基准并控制变更。如压测、演练、扩容是各部门的共性需求，但秒级监控只在超0级系统上实现

即可，是个性化需求。即使对于个性化需求，各部门的具体要求也是不一样的，如资源扩容，有的需要大内存、大硬盘机器，有的需要计算能力强的机器，绝大部分系统使用扩容的虚拟机，但个别部门只能使用物理机。

◆ **时间**。自6月1日起便进入了618期间，所有的系统准备工作必须在此之前完成。无论准备情况怎么样，6月18日当天都将出现流量峰值，这个里程碑时间节点是无法更改的。

◆ **成本**。服务器、备战物资等均有预算限制，需要进行较为准确的成本估算。

◆ **人力**。618之前，业务类项目的压力很大，可能会针对友商的促销策略提出有针对性的新业务需求，临时占用618技术备战的人力资源。

◆ **沟通**。因项目的干系人和干系部门众多，所以沟通成本较高。以组织一场全链路压测为例，项目经理需要与几十个系统的研发负责人、功能测试负责人、性能测试负责人沟通并协调压测量级、压测时间、进度安排、应急处理机制等，不能疏漏任何一个环节。因此，如何管理沟通、控制沟通就变得极为重要。

◆ **风险**。高流量情况下的系统应对属于世界性难题，需要从项目的角度帮助项目组做好风险识别与消除。如在组织全链路压测前，召集重要系统负责人、架构师参加准备会，通过头脑风暴的方式识别风险。

◆ **质量**。备战最大的困难在于如何做好零点峰值的稳定性保障，这个问题是被全世界技术界公认的技术难题，需要将漏洞、隐患、风险控制在可控范围内，确保系统的稳定。

◆ **采购**。由于服务器等硬件设备有采购周期，为确保服务器按期到货，因此需要提前相当长的一段时间进行采购，但提前时间越长、采购计划与实际需求可能差距越大，是无法避免的矛盾。

◆ **干系人**。项目组人员超过2000人，项目干系人超过4000人，这样一个量级的规模，对项目干系人的管理和控制挑战巨大。

◆ **整合**。在项目计划开发、项目计划实施及项目综合变更控制三个过程中，三个过程彼此相互影响，项目经理需要经常在相互冲突的目标或可选择的目标中权衡得失。

8.1.2 项目经理能力要求

近年来，PMI（Project Management Institute，项目管理协会）提出了项目管理人才三角理论，如图8-1所示。

1. 项目管理专业技能

项目、项目集和项目组合中特定领域相关的知识、技能与行为，包括项目管理专业知识和技能，同时加上业务领域的专业知识，可以称之为“硬实力”，是成为一个优秀项目经理必要但不充分条件。

2. 领导力

对企业完成事业目标有帮助的领导方面及跨领域的知识、技能与行为，可以称之为“软技能”，通俗



图8-1 项目管理人才三角理论



来说包括人际交往能力、认知能力、学习能力、管理能力、解决问题的能力、影响力以及情商等。

3.战略与商业管理

为提高工作表现以及更好地传递工作成果的相关行业和组织的知识与专业技能。作为项目经理，要懂业务，要有商业敏锐度，要看到并能够挖掘项目的意义和价值，厘清用户真正的需求和痛点，这一条特别符合由多个项目组成的项目集或项目组合的管理。

此外，对项目经理性格的要求是开朗、乐于沟通和分享，对618项目来说，还需要耐心和细心。

8.1.3 618项目经理需要承担的角色

要做好618的项目管理，从能力上必须满足PMI人才三角理论的要求，同时必须承担起以下角色：

- ◆ **桥梁。**项目经理是干系人沟通的桥梁，处在沟通中最核心的位置，需要保持信息畅通、准确和权威，项目经理在听、读、说、写四个方面都需具备极强的能力。

- ◆ **消防员。**出现问题要冲在前面，迅速协调资源解决问题。

- ◆ **发动机。**推进问题解决，效果及时反馈，与计划有偏差时能及时调整。

- ◆ **管家。**618项目事无巨细，既涉及各个团队之前的沟通协调，也要做好后勤保障、文化宣传、会议组织等工作。618项目管理的琐碎事宜就体现在这些地方。

- ◆ **决策中心。**项目经理根据掌握的项目信息进行项目决策或辅助总指挥进行决策，并愿意承担由此带来的风险。

- ◆ **支持中心。**除完成正常项目范围内的工作以外，还需要给其他项目或部门提供支持。

- ◆ **鼓励师。**组织动员会、慰问活动、战前鼓励及战后激励。

- ◆ **导师。**每次618项目经理都会带领1~2名新人，在完成备战项目管理任务的同时，传递京东文化价值观，促使新人快速成长。

8.2 618项目管理实践

8.2.1 启动阶段

1.每一位成员都很重要

在618项目的启动初期，项目组吸纳各个体系、各个部门中经验丰富的研发同事或项目经理加入项目管理/接口小组。项目组经过多次的沟通讨论，了解各团队的参与度和关注点，由主项目经理进行总体组织协调推进的工作。

由于大家的角色不同，每个人的需求也各有不同。例如，备战总指挥、重要系统负责人、各团队架构师往往比较关注全链路压测计划、全流程故障演练计划、联合军演计划等内容的实施，而各团队研发同事往往对各自系统的改造、优化、调整、升级工作以及服务

器支持情况比较关心。项目经理需要经过详细的调查与分析，了解每个人的需求，确定优先级和重要程度，提前解决大家关切的问题。

2.启动会

一个好的启动会对项目的开局至关重要。启动会邀请重要的领导或干系人出席，一方面可以为项目背书，另外一方面正式而庄重的启动会能显著增加仪式感和参与感，让项目显得更有意义、更重要，为项目的推进奠定良好的基础。

在618的3个月前，商城技术委员会对618备战项目进行讨论和规划。之后，经过紧密筹备，项目启动会如期而至，如图8-2所示。启动会由项目的总协调人马松主持，邀请集团各研发团队多位副总裁、系统负责人、核心研发人员和行政人员等参加。会上明确了项目总体目标、组织架构、推进形式及总指挥、项目经理人选。



图8-2 项目启动会

8.2.2 规划阶段

1.收集并分析干系人需求

通过例会、访谈、汇报、专题沟通等形式，收集备战总指挥、研发高管、各团队开发人员及采购事业部相关人员的需求，并通过问题跟踪机制对干系人需求进行持续跟踪。

2.做好时间规划

618备战项目的一个显著特点是有一个无法延期的里程碑时间节点，即从6月1日618开始，所有资源、活动必须准备完毕。项目重要里程碑时间节点如图8-3所示。

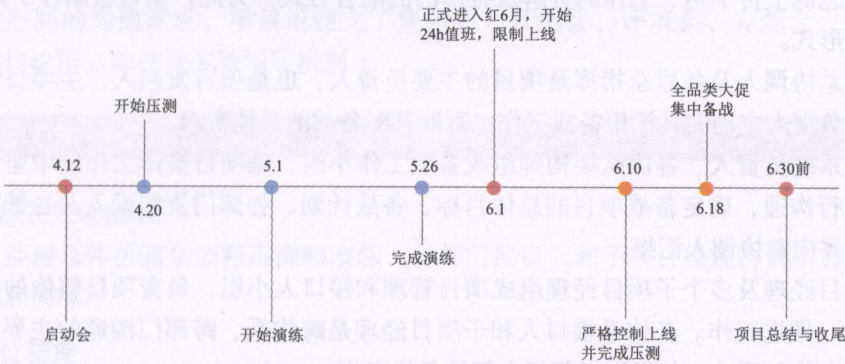


图8-3 项目重要里程碑

为获得备战项目实施的最高效率，需要对活动进行排序，确定各种活动之间的依赖关系和逻辑关系，估算每个活动所需资源及对活动进行排序。影响活动排序的关键要素包括



各活动之间的依赖关系、活动之间的人为依赖关系及外部依赖关系。例如压测演练的逻辑顺序为：先压测，提交服务器需求，到货，服务器扩容，再压测，先小团队军演，再联合军演。在确定逻辑顺序后，具体重要里程碑时间节点也就较为明确，项目的进度计划即可依据这些里程碑节点制定。多数情况下，多个活动是可以并行进行的。但部分活动只能串行，对于串行的活动，在确定每个活动的工期之后，把有逻辑关系的活动用网络图的方式连接起来，并计算项目的最短工期，形成关键路径，如图8-4所示。

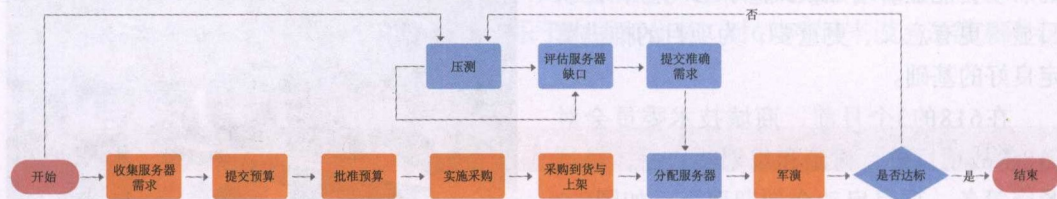


图8-4 项目的一条关键路径

3.做好资源规划

打仗讲究“兵马未动，粮草先行”，618项目也一样。服务器、网络设备以及备战物资的采购需要提前启动。

◆ **服务器、网络设备的准备。**设备的采购与到货是保证618项目成功的关键因素。为扛住618期间流量洪峰的压力，除了需要优化正常的系统之外，还额外需要大量的各种类型服务器设备，包括应用CDN设备、存储设备、大内存设备、大数据设备、基本应用服务器、计算型设备及数据库设备。由于大部分服务器到货周期较长，为保证服务器按时到位，成立设备管理子项目，确保急需设备按时到货。

◆ **备战物资的准备。**备战物资包括加班餐饮、文化纪念品、桌椅、活动服务等，基本集中在6月1日至6月19日。需确定活动主题并进行相应的物资采购。

4.确定组织形式

本项目干系人和干系部门众多，需要建立一种比较有效的组织方式，保证备战计划的执行、信息的上传下达、工作的分解及各部门的配合方式。为此，项目组制订了如图8-5所示的组织形式。

项目总协调人及备战总指挥是项目的主要负责人，也是项目发起人，主项目经理组织重要系统负责人主动向其汇报备战工作，听取其对备战的总体要求。

重要系统负责人、各团队架构师组成备战工作小组，是项目整体工作的中坚力量。需要集中进行沟通，确定备战项目的总体目标、备战计划、各部门资源投入及各部门备战的重要事项并向总协调人汇报。

主项目经理及多个子项目经理组成项目管理和接口人小组，负责项目整体的沟通、协调、组织、推进工作。各体系接口人和子项目经理是跨体系、跨部门沟通的主要渠道，负责信息的上传和下达，并组织本部门内部的备战工作。

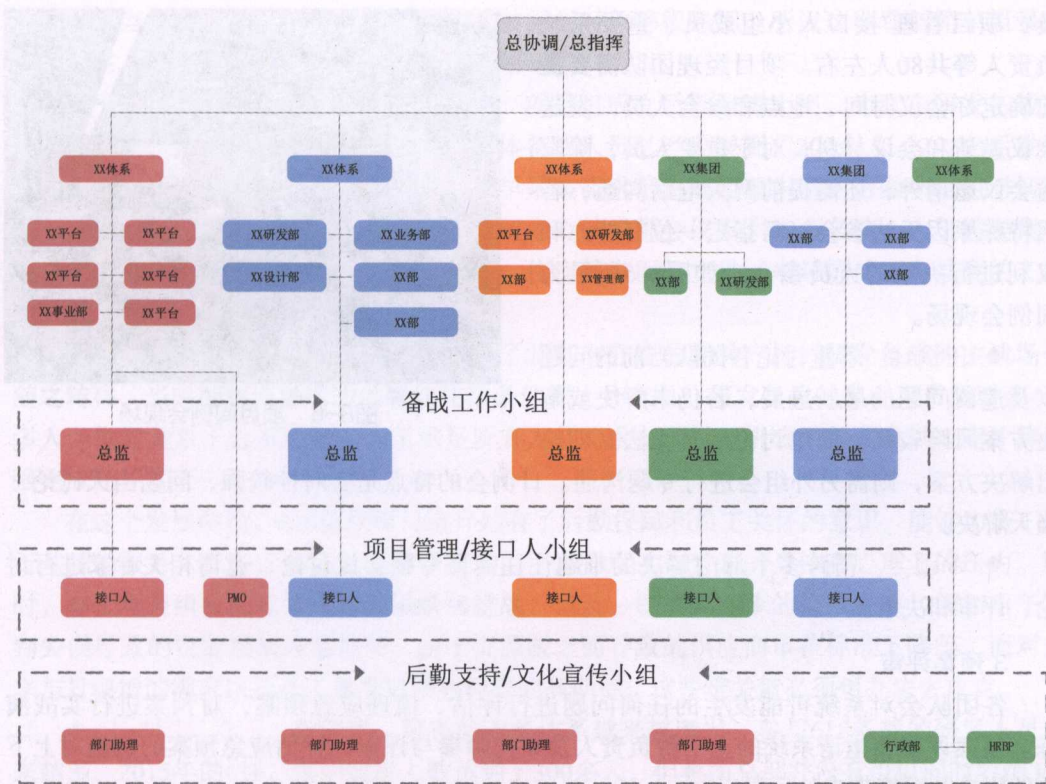


图8-5 项目组织架构

后勤支持/文化宣传小组负责备战的后勤支持与文化宣传工作，由子项目经理、各部门助理及集团行政人员组成。

主项目经理总体上负责协调各个小组以及各体系研发团队有序配合，步调一致地完成备战工作。

5.建立沟通机制

依据不同的沟通需求，项目组建立了周例会、日例会、评审会、专题会、咚咚讨论群、微信讨论群、邮件等多种沟通机制。

8.2.3 执行阶段

1.组织压测和演练

项目经理总体协调全流程压测和演练，各部门接口人和子项目经理协调本部门内部的单体压测和演练。

2.会议管理

◆ **周例会。**项目组每周组织召开一次全体周例会。周例会在备战过程中有着极其重要的作用，它既是讨论、决策和解决问题的渠道，也是同步备战信息、问题解决进展和交流备战经验的方式。周例会参会人员较多，包括各研发部门副总裁、备战工作小组成员



员、项目管理/接口人小组成员、重要系统负责人等共80人左右。项目经理团队需要提前确定好会议时间、地点和参会人员，发送会议邀请和会议材料；对于重要人员，除发送会议邀请外，还需提前两天电话沟通，若因特殊原因无法参会，需指定一位有能力和权利进行决策的人员参会。如图8-6所示为周例会现场。

◆ **日例会。**着重讨论各团队当前的问题以及遗留问题的最新进展，若仍未解决或解决方案风险较高，简单讨论后若无法立即给出解决方案，则需另外组会进行专题沟通。日例会的特点是实时性较强，问题当天讨论，当天解决。

◆ **专题会。**需跨多个部门解决的难题往往需要专题会议讨论，邀请相关专家进行讨论、评审和决策。

3. 预案评审

各团队会对系统可能发生的任何问题进行评估，梳理应急预案，对预案进行实战演练。预案评审会邀请系统的上下游负责人及架构师参与评审，评估应急预案的实施对上下游的影响及可能带来的后果。

4. 应急响应

为快速响应可能存在的系统问题，6月1日起开始执行24h现场集中值班制。0级及1级系统安排核心研发人员在集中备战场地值班，架构师团队、项目管理部每天各安排一人值夜班。

5. 集中备战

6月17日至6月19日，所有重要系统负责人、测试人员近千人在固定的场地集中值班，称为集中作战中心，以便在应急情况下方便与上下游系统沟通。

6. 后勤保障

618备战期间，对于员工的体力和精力都是巨大的考验，所以做好备战的后勤保障工作，是保持源源不断战斗力的必要条件。后勤支持/文化宣传小组重点围绕着员工办公、吃、住、行这四个方面的需求给予支持，力求让员工在最舒适的环境中，精神饱满地备战618。

后勤保障模式优化。在历年备战慢慢升级的过程中，后勤工作也从最初的被动救火式保障，升级到如今的主动服务式保障。

◆ **初级救火阶段。**早年间所谓的618备战，其实只有几名研发的核心人员24h值班。所谓的后勤保障，也就是每到了吃饭时间，助理订一些快餐，买几碗泡面就算解决了吃的问题。而住呢，要么是员工自己定酒店，后续走报销流程，要么是直接在工位上简单小憩。所有的后勤需求都是员工临时突然提出，然后助理们临时为大家寻求解决方案的救火式服



图8-6 项目周例会现场

务。由于当时的备战人员少，服务对象和被服务对象都很熟识，所以这种看似简单粗暴的供需关系，却恰恰能够快速响应，快速解决问题。

◆ **基础服务阶段。**到了2013年的618，项目组成立后勤保障小组，为京东研发备战人员保驾护航，扫平后顾之忧。这时候，整体备战人员已经发展到了200余人。2~3名后勤保障人员已无法快速响应所有人的需求，提早采集需求信息和经验丰富的后勤项目组对情况的预判，变得尤为重要。在北京市北辰世纪中心的成都会议室，40多名核心人员需要集中在一起。后勤保障小组提前做好所有餐食、住宿等方面的安排，来对抗员工高强度的工作压力。

到了2015年，大部分的研发人员搬到了北京市亦庄的朝林广场，研发备战的主战场也随之转移。当时朝林广场周边的餐饮住宿条件都十分有限，研发备战的集中人员激增到400多人，供需关系十分不平衡。为了满足所有人的需求，项目组拼凑了很多家供应商才算勉强完成任务。

在这个发展阶段，618备战项目组开始有了后勤保障和员工关怀的意识，能够为员工进行基础的服务，在一定程度上缓解了员工由于高强度工作带来的身体和心理上的压力。同时，整个项目组也面临了很多的困难和挑战，以及一些需要变革的待改进项。例如由于预判失误导致的资源短缺或是浪费，由于资源匮乏而导致的供应商审核标准下降等。面对未来与日俱增的需求以及员工本身对服务标准的高期待，这些难关就必须想办法去攻克。

◆ **主动服务阶段。**近两年，京东的业务体系越来越庞大，参与618备战的研发人员越来越多，2015年的618，集中值班人数达到了300多人，但集团总部并没有可以同时容纳300人的会议室。项目组与行政部沟通，对现有会议室进行电力系统改造，在不影响会议室正常运转的前提下，6月份之前务必完工，所以一切改造都需要一部分一部分地进行，并且全部作业都要在非工作时段进行。考虑到集中备战规模越来越大，会议室的电力系统改造预留了一定的容量，并顺利支撑了2017年的618期间900多人规模的集中备战，如图8-7所示。



图8-7 部分集中备战场地

京东研发备战总指挥部“京东视界”的软硬件升级也是作战室整体改造的重中之重。供应商的选择，装修方案的敲定，环保、品质的把控，装修的工期一缩再缩，都离不开项目组的积极协调及多方高效合作。



配套服务升级

◆ **吃好喝好。**在后勤保障工作里，吃饭的问题永远都是考虑最多的问题。在炎热的夏季，如何让员工吃得更安全、更健康、更美味、更便捷是项目组布置相关工作的首要考虑的。

在之前的经验中，由于工作地点内部提供餐饮的场所少，研发人员用餐的时间紧张，人数众多，因此对于餐食的供应，解决方案是外部订餐后发放到大家手中。在这个过程中，风险点极多：配送损耗，天气导致的食物变质概率提高，口味不适导致的员工宁愿自行用餐。这每一点都造成了极大的浪费，并且降低了员工的满意度。针对这一问题，面对京东总部内部的资源，项目组决定利用专属餐券的方式为员工提供餐饮服务，开辟出研发专属用餐区域以满足研发人员用餐时间紧张的难题，并延长餐厅服务时间，开通专属绿色窗口。根据后期结算数据显示，利用餐券的供应方式，扣除掉人员增长的系数后，能够节省大约30%的成本，还释放出了大量后勤保障的人力资源。

◆ **“享”睡就睡。**睡眠是修复疲累最好的良药，为了能够让员工更好的休息，项目组将需求整理为了“住宿需求”和“小憩需求”两个方面。

在住宿需求上，后勤小组按照体系与部门的划分，分别和亦庄的多个协议酒店协调住宿资源，实名到人，对公结算。员工只需要凭身份证就可以入住酒店，不需要预付任何的费用。在小憩需求上，项目组分散地封闭了若干会议室，用行军床和毛毯临时打造了一个个集中小憩休息区，分别为男、女员工供应。小型休息室提供简单的一次性清洁用品、驱蚊用具以及眼罩，让员工在公司也能在短时间内进入到一个舒适的睡眠环境中。

◆ **出行无忧。**针对618备战，在出行方面，后勤小组也从班车、打车、共享汽车和共享单车四种途径为员工尽可能地提供便利。

班车方面，除了延后了常规两组班车的发车时间外，还增加了晚间从公司到协议酒店，公司到周围住宅区的摆渡车。从22:30到凌晨03:30，每小时一个班次。解决了夜间近距离打车难的问题。

打车方面，员工内部软件“京ME”完美地接入了滴滴打车平台，员工无须自己先行垫付车款后再走报销流程，平台会直接和公司进行结算，极大地提升了用户体验和效率。

除了以上两种形式，在出行方面，项目组积极和供应商协调，增加共享汽车和共享单车的布控数量，给予员工更多的出行选择。

◆ **能量加油站。**618期间，邀请按摩师来给大家放松身体、在作战室外定点发放消暑绿豆汤、利用京东无人车发放补充能量包等。

除了基础的保障，618后勤项目组还力求从更多的方面补充员工关怀工作，期望能够为员工带来更多的幸福感和满足感，让大家在轻松愉快的氛围下快乐备战。

通过提供各种形式的后勤保障，解决了备战人员的后顾之忧，让大家全心铺在系统保障和支持上。

7.宣传激励

在618前夕，后勤支持/文化宣传小组也会通过组织各种文化宣传活动和慰问活动激发干系人参与的热情，缓解紧张的气氛。以下通过简单介绍几个有代表性的活动，来让大家

一同感受京东618的火热气氛。

宣传标语。口号和标语向来是做宣传工作的首要选择。合辙押韵、朗朗上口的口号，通过条幅、旗帜、易拉宝等形式展现，能够快速点燃618备战的热烈气氛，增强全体研发人员的归属感和使命感。每次备战口号的设计灵感，都源于日常工作，又高于日常工作。

618慰问

◆ **高管慰问。**无论是什么工作，京东的高管永远都并肩和兄弟们站在一起，他们时时刻刻惦念着在后方给予京东莫大支撑力的兄弟们，总是要带着满满的诚意犒劳慰问，如图8-8所示为618当天高管慰问加班员工。



图8-8 618当天高管慰问加班员工

◆ **主题慰问。**每年的主题慰问往往是茶余饭后最好的谈资。无论是美丽与智慧并存的产品经理Cosplay主题（图8-9），还是乖萌与暖心的京宝贝父亲节主题（图8-10），都为618紧张的气氛带来一丝轻松。



图8-9 2016年618 Cosplay主题慰问

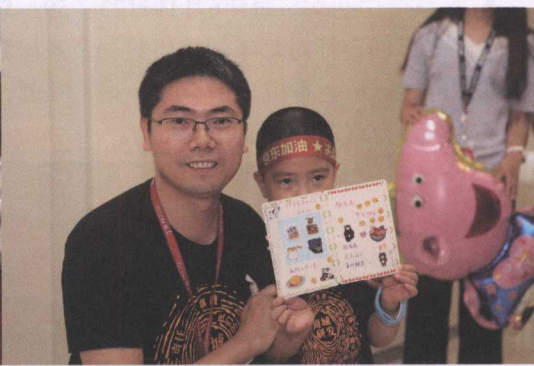


图8-10 2017年618父亲节主题慰问

618创意展示。618期间，后勤支持/文化宣传小组组织和京东备战主题相关的创意展示。自2016年的《京东all day》响彻京东总部大楼之后，2017年618的《京·东游记》更是唱出了研发人的心声，流露出了员工对工作、对生活的热爱，营造了炙热的备战氛围，如图8-11所示。



图8-11 2017年618快闪合唱团

◆ 传说中的“召唤神龙JOY”。收集癖每年最期待的，大概就是618备战期间的特殊限量版纪念品了，如图8-12所示。每一份纪念品都是由JDC根据每年的特色精心设计的。京东流传着一个神秘的传说，据说集齐7年的纪念品便能够“召唤神龙JOY”，实现一个愿望。



图8-12 部分618纪念品

◆ 零点庆功活动。每年6月19日23:59:59，所有的销售数字会在指挥中心的大屏幕上停止。老板们会针对每年的销售数据及系统数据进行复盘，随后便是与员工们的庆功狂欢，如图8-13所示。香槟、蛋糕、炸鸡、啤酒，平稳的系统 and 傲人的销售数据，这一切都成为了京东人为京东庆生最好的礼物。



图8-13 零点庆功活动

618备战的文化宣传工作，最大的特点就是“从群众中来，到群众中去。”因为每一位京东人都是一个个鲜明的标签，时刻展现着京东最核心的战斗力量。每一个笑容都值得记录，每一句话语都值得宣传。相信随着京东影响力的日益增强，京东的文化输出也会越来越

越成功。

8.2.4 监控阶段

本项目的监控流程总体上遵循商城研发体系PMO既有的流程与管理办法。在本项目中，项目监控过程中，项目经理的监控方式包括组织周例会、日例会、应急预案评审以及对重要问题跟踪，对项目组成员的项目任务及工时进行审核，对于出现的风险及问题制定评审解决方案并跟踪监控措施。项目总协调、总指挥以及备战工作小组通过参加周例会、日例会、应急预案评审会、汇报会等形式对项目进展情况进行监控。监控过程的主要产出物包括会议纪要、任务审批记录、变更审批记录、项目周报等。项目监控流程与管理方法如图8-14所示。

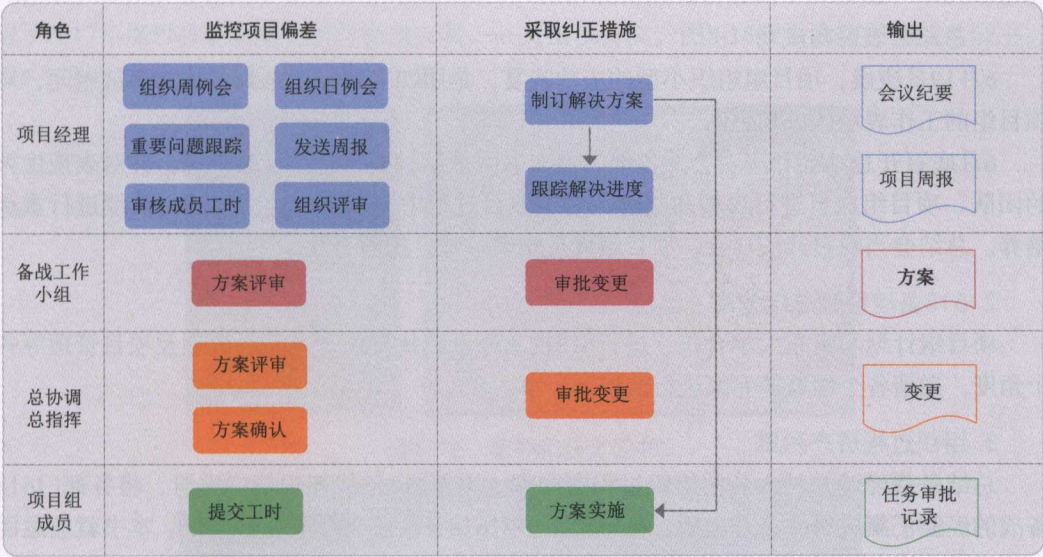


图8-14 项目监控流程与管理方法

1. 建立问题跟踪机制，做好进度跟踪和控制

为监控项目进度与项目计划的偏差程度，项目组采取了以下问题跟踪机制：在周例会、日例会及其他渠道反馈的需跨部门协调，且无法立即解决的问题，项目经理记录到问题跟踪表中，限定解决时间，并分配责任人和责任部门；跟进人每日跟进问题解决进度，协调资源并推进问题解决，在每周五向所有干系人发送的项目周报中附上问题跟踪表，同步最新解决进展。

2. 问题重要紧急程度不同，触达渠道也不同

根据问题或风险的重要紧急程度，建立不同的反馈渠道，常规问题反馈至备战接口人群，抄送备战工作小组；重要问题反馈至备战工作小组群组，抄送主管副总裁、备战总指挥及项目经理，同时在备战咚咚群中发送消息；紧急问题必须第一时间发送至微信“0级系统负责人群”，相关负责人会第一时间关注并尽快解决；重要且紧急的问题除第一时间发



送至微信“0级系统负责人群”紧急处理外，还需在问题解决之后，把问题描述、原因、解决方案发送至邮件发送相关干系人，并抄送主管副总裁、备战总指挥、项目经理、备战工作小组及项目管理/接口人小组。

3. 利用项目管理系统（PMP），辅助做好项目管控

京东自研的项目管理系统——京东研发管理平台（PMP3.0），从项目基本信息、工时填报及审批、风险和问题管理、干系人管理、项目进度管理、变更管理、成本核算等方面，对项目全方位进行管控，尤其是支持基于任务填报工时、审核工时，成本估算精细到具体任务，对项目进行高效率的指导和控制。

8.2.5 收尾阶段

1. 总结、表彰与庆功

6月19日凌晨，项目组组织小型的庆功晚宴，集团CEO刘强东来到备战指挥部慰问，对项目组的工作表示认可和感谢。

6月底召开总结会和庆功会，各部门提交书面总结材料，总结经验教训，并对表现优秀的团队、项目组成员进行表彰和奖励，对于项目过程中涌现出的优秀人才，后续进行重点培养。总结会当晚召开庆功会，项目组成员欢聚一堂，把酒言欢。

2. 618备战经验总结分享

项目组计划与京东大学合作，进行618技术分享，从架构、产品、管理及项目管理等各个角度，邀请各个领域的专家进行演讲和分享。

3. 组织过程资产积累

总结备战经验并纳入公司组织过程资产库，更新618备战知识精华手册，将各部门618备战的经验汇集成册并公开出版，分享给更多对618备战感兴趣的业界同仁。本书就应运而生，希望对读者有所帮助。

8.3 项目成果

通过618技术备战，不但支撑了集团的销售目标，保证系统安全平稳。更重要的是在这个过程中，为人工智能、机器学习、大数据、弹性云等新技术提供了应用和验证的场景，进一步夯实了系统的基础，留下了极其宝贵的组织过程资产。在备战过程中培养了很多技术一流并符合京东价值观的技术人才。同时在社会 responsibility 方面做出了贡献：

◆ 京东无人技术在618全面落地运营，无人配送机器人在中国人民大学送出日常配送运营第一单，京东无人机常态化运营在西安、宿迁开启，无人仓技术在武汉亚洲一号仓库投入运营，无人客服机器人代替了大量的人工客服的工作量。以上新技术的大量采用，使得成本降低，效率提高，树立了行业标杆，推动了物流行业的技术进步和转型升级。

◆ 升级风控技术，把网络黄牛挡在订单之外，把实惠传递到普通用户手中。

- ◆ 升级人工智能图像识别技术，在京东站内没有问题图片，净化了网络环境。
- ◆ 通过技术手段主动识别并防止刷单现象，给用户展示最准确的销量和真实的评价数据，树立良好的行业风气。
- ◆ 全面应用大数据在数百万的商品中识别水货和假货，确保不卖出任何一件水货和假货。
- ◆ 采购了更节能的服务器，升级现有弹性云技术，实验减少CPU空转的新技术，有望大幅减少服务器功耗，实验成功即可向社会开放。
- ◆ 项目管理团队综合运用项目组合、项目集管理理论、项目管理工具、项目管理经验及组织过程资产，并结合京东实际情况，在项目的各个阶段做好沟通、组织、协调、推进工作，做到事前有规划，事中有监控，事后有总结。通过卓有成效的项目管理，项目完美支撑了618的大促活动，并受到了国内知名项目管理专家的肯定。在2017年9月23日召开的PMI（中国）项目管理年会上，京东集团618大促研发备战项目受到了评委的一致认可，获得了2017年度PMI（中国）项目管理大奖——杰出项目奖。图8-15所示为奖杯和获奖证书，图8-16所示为项目经理在颁奖现场合影。



图8-15 奖杯和获奖证书



图8-16 项目经理在颁奖现场合影

互联网行业推崇产品为王，项目组织一般都是弱矩阵或平衡矩阵形式，项目经理有责无权，要推功揽过。在传统软件行业，很多公司的项目经理由研发经理兼任；在互联网行



业，很多公司的项目经理由产品经理兼任，项目经理在夹缝中生存，相对来说是弱势群体。但不管怎么样，项目经理都至少是名义上的项目负责人，是项目沟通的桥梁，是项目协调的核心力量，是组织者更是推动者，甚至经常还是决策者。所以，最关键的不是组织授权给项目经理什么权利，而是自己能承担起什么样的角色，发挥什么作用，起到哪些正面的推动作用和影响。尤其是涉及非常多的系统、团队和干系人的项目，研发经理和产品经理在专业领域的知识和能力无法满足项目推进的要求，只能通过项目经理在项目管理方面的知识和经验，在沟通、组织、协调、推进等通用能力以及在业务和系统的方面的洞察力，才能真正掌控项目，体现项目经理和项目管理的价值。

回顾历年的618备战，从最初的不知所措、摸着石头过河，到现在的各团队胸有成竹、井然有序。更重要的是，在备战过程中，培养了一批又一批技术人才，传承了能打硬仗、能挑重担、团队配合、团结共赢的研发精神。这种精神，也是京东下个12年实现在技术引领下开拓创新所必不可少的财富。



科导：商城研发部PMO项目经理，总体负责618技术备战的项目管理工作。

瞳瞳：负责后勤保障与文化宣传工作。

参考文献

- [1] 项亮. 推荐系统实践 [M]. 北京: 人民邮电出版社, 2012.
- [2] Dietmar Jannach, Markus Zanker, Alexander Felfernig, et al. 推荐系统 [M]. 蒋凡, 译. 北京: 人民邮电出版社, 2013.
- [3] Ralph Kimball, Margy Ross. 数据仓库工具箱——维度建模权威指南. [M]. 王念滨, 周连科, 韦正现, 译. 3版. 北京: 清华大学出版社, 2015.
- [4] 王磊. 微服务架构与实践 [M]. 北京: 电子工业出版社, 2016.
- [5] 盛骤, 谢式千, 潘承毅. 概率论与数理统计 [M]. 北京: 高等教育出版社, 2002.
- [6] Peter Flach. 机器学习 [M]. 段菲, 译. 北京: 人民邮电出版社, 2016.
- [7] 张捷, 朱海萍, 胡州明, 等. 基于系统自身业务实现的分库分表策略 [J]. 自然科学: 全文版, 2016(9): 00249-00250.
- [8] 陈传波, 祝中涛. RSA算法应用及实现细节 [J]. 计算机工程与科学, 2006, 28(9): 13-14.
- [9] 何明星, 林昊. AES算法原理及其实现 [J]. 计算机应用研究, 2002, 19(12): 61-63.

618对于京东来说是一个大考，也是一个练兵场，我们在其中获得了不少的经验心得，锻炼出来一支能征善战的研发队伍，也有着很多人刻骨铭心的教训。

在阅读本书后，如能让您对电商的最新技术成就和发展趋势有新的了解，将是对作者们的最佳回报。

与君共勉！



扫描二维码
关注京东商城研发

技术需要积累，需要沉心静气努力钻研，但同时也需要同行之间的相互交流。京东作为国内领先的互联网公司，技术团队有丰富的项目实操经验，能够在这样一本书里将京东最重要的“618”项目备战经验，尤其是一些技术研发细节如系统架构信息总结分享出来实属不易，值得研发同行一读。

★周鸿祎★
360公司董事长兼CEO

618是京东的生日，更是广大网民的年度购物狂欢节。2017年的618期间，京东的下单金额累计达1199亿元。这个数字的背后，绝对是一场看不见硝烟的IT技术人集体大会战。

作为国内电商的领先者，京东在技术实力和创新性上，都始终站在浪潮前沿，618更是其综合实力的集中呈现。我个人非常推荐这本书，她从一个个订单的诞生开始，到移动体验和智慧服务的实现，再到智能物流以及强健的基础支撑体系的打造，如庖丁解牛般地逐层分析，让读者既能站在整体架构上一窥全局，又能从实战中得以了解技术内幕，是一本难得的电商类技术图书佳作。

★蒋涛★
CSDN&极客邦创始人

大数据时代，数据驱动的科学决策正成为企业资源优化配置的利器。京东是目前国内电商中最重视数据技术和商业智能结合的公司之一，书中也非常详尽地介绍了京东在618备战前后是如何运用这些数据来科学决策的，对于电商从业者乃至所有希望获得商业智能的公司都是非常好的借鉴。

★叶荫宇★
美国斯坦福大学管理科学与工程系及计算数学工程研究院杰出终身教授
K.T.Li（李国鼎）首席教授（Chair Professor）

拿到这本书的样稿时，我脑海中第一闪现的，是刘强东在京东十二周年的庆典大会上提到的京东发展战略，他说“未来的十二年，京东将全面技术化，大力发展人工智能和机器人自动化技术。”我想这本书是对京东技术战略最好的呼应。

技术能否落地一直是很多公司的大难题，但京东研发团队做到了。短短几年内，他们快速成长，很好地支撑了业务的全面发展。这本书详细阐述了618这个年度营销重头戏背后研发团队的努力，对于从事电商工作的技术人员来说，是个不错的参考。

★霍泰稳★
InfoQ中国创始人、极客邦科技创始人兼CEO



博文视点Broadview



京东图书
—JD.COM—



策划编辑：孙学瑛 康旭 宋亚东
责任编辑：孙学瑛
封面设计：李玲

特约监制：杨海峰 特约策划：吴迪

上架建议：电子商务 / 畅销书

ISBN 978-7-121-32730-8



定价：99.00元